ioProgrammo Anno XI - N°6 (115)

FILE SHARING: CREA IL TUO MULO DA BATTAGLIA • METTI WINDOWS IN SICUREZZA • WINDOWS PRESENTATION FOUNDATION

EDIZIONI MASTER

ANTEPRIMA: BORLAND DELPHI PER PHP

TUTTE LE NOVITÀ DELL'AMBIENTE RAD CHE TI FA CREARE LE TUE PAGINE PHP SEMPLICEMENTE DISEGNANDOLE!

VERSIONE PLUS RIVISTA+LIBRO+CD €9,90

VERSIONE STANDARD ✔ RIVISTA+CD €6,90

PER ESPERTI E PRINCIPIANTI

Poste Italiane S.p.A Spedizione in A.P. • D.L. 353/2003 (conv.in L.27/02/2004 n.46) art.1 comma 2 DCB ROMA Periodicità mensile • GIUGNO 2007 • ANNO XI, N.6 (115)

FILE SHARING

Crea il tuo MULO DA BATTAGLIA

Svelato il codice di eMule! Ecco cosa si nasconde dietro al software più usato in rete

- ✔ **ARCHITETTURA** la rete edonkey e il formato dei messaggi TCP/IP
- ✔ **CONNESSIONE** le routine per effettuare il login con i server
- ✔ **RICERCA** i comandi per scovare musica, film, software e tutto il resto
- ✔ **DOWNLOAD/UPLOAD** come scaricare i file, salvarli sul disco e condividerli con gli altri

INSTALLAZIONI di Vista™ PERSONALIZZATE

Con WAIK crei un'immagine del sistema con il software e la grafica di setup che vuoi e la distribuisci!

JAVASCRIPT

INTRODUZIONE A DOJO
Ecco gli esempi per aggiungere al tuo sito pagine interattive. Anche per chi non è un esperto...

GRAFICA CON GLI SFONDI DINAMICI
Sfruttiamo i fogli di stile per creare "gradienti" al volo senza usare photoshop

UN WORD PROCESSOR NEL BROWSER
Ecco come creare form che consentono una raffinata formattazione del testo

msdn WEBCAST 3 NUOVI VIDEO

• PERCHÉ SVILUPPARE PENSANDO DA SUBITO ALLA SICUREZZA
• LA VALIDAZIONE DELL'INPUT • AUTENTICAZIONE E AUTORIZZAZIONE

METTI WINDOWS IN SICUREZZA

Alla scoperta del Security Configuration Manager: il tool per trovare i buchi nella tua installazione e risolverli

DISEGNA SUBITO CON .NET 3.0

Con Windows Presentation Foundation tutto è vettoriale. Ecco la guida per realizzare facilmente la tua prima applicazione con grafica animata

DATABASE

IMPORTA LE TABELLE DA UN DB ALL'ALTRO
Creiamo script personalizzati per importare ed esportare dati e strutture senza problemi

WEB

LE LIBRERIE USATE DA YAHOO
Introduzione a YUI: il framework JavaScript usato dai più grandi

JAVA

ARRIVA DERBY IL DB EMBEDDED
Sfrutta al massimo il nuovo database incluso in Java6

VISUAL BASIC

GESTIONE DEL FORMATO XML
Creiamo classi al volo partendo da un file di definizione

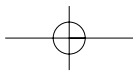
C++

TROVIAMO TUTTO CON LE REGULAR EXPRESSION
Ricerche di stringhe superveloci grazie alla Regex

ALGORITMI: Sfruttiamo il calcolo della mediana per creare strutture ordinate risparmiando tempo e cicli macchina

EDIZIONI MASTER www.edmaster.it

70115
9 771128 594009

2-05-2007 15:25:18

Anno XI - N.ro 06 (115) - Giugno 2007 - Periodicità: Mensile
Reg. Trib. di CS al n.ro 593 del 11 Febbraio 1997
Cod. ISSN 1128-594X
E-mail: ioprogrammo@edmaster.it
http://www.edmaster.it/ioprogrammo
http://www.ioprogrammo.it

Direttore Editoriale: Massimo Sesti
Direttore Responsabile: Massimo Sesti
Responsabile Editoriale: Gianmarco Bruni
Vice Publisher: Paolo Soldan
Redazione: Fabio Farnesi
Collaboratori: R. Allegra, D. De Michelis, F. Grimaldi, E. Viale, V. Vessia, A. Pelleriti, F. Smelzo, C Scuderi, G. Malaga, F. Fortino, V. Arconzo
Segreteria di Redazione: Rossana Scarcelli

Realizzazione grafica: Cromatika S.r.l.
Art Director: Paolo Cristiano
Responsabile grafico di progetto: Salvatore Vuono
Coordinamento tecnico: Giancarlo Sicilia
Illustrazioni: M. Veltri
Impaginazione elettronica: Francesco Cospite, Lisa Orrico, Nuccia Marra, Luigi Ferraro

Realizzazione Multimediale: SET S.r.l.
Realizzazione CD-Rom: Paolo Iacona

Pubblicità: Master Advertising s.r.l.
Via C. Correnti, 1 - 20123 Milano
Tel. 02 831212 - Fax 02 83121207
e-mail advertising@edmaster.it
Sales Director: Max Scortegagna
Segreteria Ufficio Vendite: Daisy Zonato

Editore: Edizioni Master S.p.a.
Sede di Milano: Via Ariberto, 24 - 20123 Milano
Sede di Rende: C.da Lecco, zona industriale - 87036 Rende (CS)
Presidente e Amministratore Delegato: Massimo Sesti
Direttore Generale: Massimo Rizzo

ABBONAMENTO E ARRETRATI

ITALIA: Abbonamento Annuale: ioProgrammo (11 numeri) €59,90 sconto 21% sul prezzo di copertina di €75,90 · ioProgrammo con Libro (11 numeri) €75,90 sconto 30% sul prezzo di copertina di €108,90 Offerte valide fino al 30/06/07.
Costo arretrati (a copia): il doppio del prezzo di copertina + € 5.32 spese (spedizione con corriere). Prima di inviare i pagamenti, verificare la disponibilità delle copie arretrate allo 02 831212.
La richiesta contenente i Vs. dati anagrafici e il nome della rivista, dovrà essere inviata via fax allo 02 83121206, oppure via posta a EDIZIONI MASTER via C. Correnti, 1 - 20123 Milano, dopo avere effettuato il pagamento, secondo le modalità di seguito elencate:

- cc/p n.16821878 o vaglia postale (inviando copia della ricevuta del versamento insieme alla richiesta);
- assegno bancario non trasferibile (da inviarsi in busta chiusa insieme alla richiesta);
- carta di credito, circuito Visa, Cartasì, o Eurocard/Mastercard (inviando la Vs. autorizzazione, il numero di carta di credito, la data di scadenza, l'intestatario della carta e il codice CVV2, cioè le ultime 3 cifre del codice numerico riportato sul retro della carta).
- bonifico bancario intestato a Edizioni Master S.p.A. c/o BCC MEDIOCRATI S.C.AR.L. c/c 0 000 000 12000 ABI 07062 CAB 80880 CIN P (inviando copia della distinta insieme alla richiesta).

SI PREGA DI UTILIZZARE IL MODULO RICHIESTA ABBONAMENTO POSTO NELLE PAGINE INTERNE DELLA RIVISTA. L'abbonamento verrà attivato sul primo numero utile, successivo alla data della richiesta.

Sostituzioni: qualora nei prodotti fossero rinvenuti difetti o imperfezioni che ne limitassero la fruizione da parte dell'utente, è prevista la sostituzione gratuita, previo invio del materiale difettato.
La sostituzione sarà effettuata se il problema sarà riscontrato e segnalato entro e non oltre 10 giorni dalla data effettiva di acquisto in edicola e nei punti vendita autorizzati, facendo fede il timbro postale di restituzione del materiale.
Inviare il CD-Rom difettoso in busta chiusa a:
Edizioni Master - Servizio Clienti - Via C. Correnti, 1 - 20123 Milano

Servizio Abbonati:
☎ **tel. 02 831212**
@ **e-mail: servizioabbonati@edmaster.it**

Assistenza tecnica: ioprogrammo@edmaster.it
Stampa: Arti Grafiche Boccia S.p.a. Via Tiberio Felice, 7 Salerno
Stampa CD-Rom: Neotek S.r.l. - C.da Imperatore - Bisignano (CS)
Distributore esclusivo per l'Italia: Parrini & C S.p.A.
Via Vitorchiano, 81 - Roma

Finito di stampare nel mese di Maggio 2007

1 Anno di Computer Bild 2006, 1 Anno di Io Programmo in DVD 2006, 1 Anno di Linux Magazine in DVD 2006, 1 Anno di Office Magazine 2006, 1 Anno di Win Magazine in DVD 2006, Audio/Video/Foto Bild Italia, A-Team, Calcio & Scommesse, Colombo, Computer Bild Italia, Computer Games Gold, Digital Japan Magazine, Digital Music, Distretto di Polizia in DVD, DVD Magazine, DVD Magazine Films, Family DVD Games, Filmteca in DVD, Go!OnLine Internet Magazine, Home Entertainment, Horror Mania, I Film di DVD Magazine, I DVD di Win Magazine, I DVD de La Mia Barca, I Film di Idea Web, I Filmissimi in DVD, I Film di DVD Magazine, I Grandi Giochi per Pc, I Libri di Quale Computer, I Mitici all'Italiana, Idea Web, Idea Web Film, InDVD, IoProgrammo, I Tecnoplus di Win Magazine, Japan Cartoon, La mia Barca, La mia Videoteca, Le Femme Fatale del Cinema, Le Grandi Guide di Io Programmo, Linux Magazine, Magnum PI, Miami Vice in DVD, Nightmare, Office Magazine, Play Generation, Play Generation Games, Popeye, PC Junior, Quale Computer, Softline Software World, Sport Life, Supercar in DVD, Star in DVD, Video Film Collection, Win Junior, Win Magazine Giochi, Win Magazine, Le Collection.

Questo mese su ioProgrammo

NUOVE FRONTIERE

Lo ha detto Bill Gates: "non ci sono limiti allo sviluppo dell'informatica". Non arriveremo mai al punto di stallo della scienza. D'altra parte a dar ragione a Gates è la storia. Ciò che qualche anno fa sembrava pura fantascienza oggi è normalità. Così esistono dei mercati che oggi sembrano sempre sul punto di esplodere per poi assumere una posizione più defilata. Parliamo ad esempio del mercato dei device portatili. Sono ormai anni che dipingiamo questo settore come strategico. E sono ormai anni che tuttavia ci ritroviamo a sviluppare per il desktop. Allo stesso modo Internet è da sempre definita come strategica, ma alla luce dei fatti alla fine il grosso dello sviluppo sta ancora sul mercato Desktop. Eppure questi settori sottilmente si sono insinuati nella nostra vita quotidiana, talmente tanto da non poter più rappresentare un mercato in via di sviluppo. Internet è l'elettrodomestico di questo secolo, ha la stessa diffusione di un televisore o di una lavatrice. Le applicazioni che girano su Internet sono ormai alla portata di tutti. Lo stesso identico concetto è applicabile ai palmari e ai cellulari. È dunque il momento di sognare ancora. Di pensare ad applicazioni che oggi giorno non sono alla portata di nessuno. Che sfruttino totalmente l'utopia di una banda ultralarga, che facciano largo uso dell'intelligenza artificiale, che approccino mercati come la robotica e l'embedded. Dobbiamo sganciarci dal concetto che il mercato emergente è il mobile o internet e cominciare a pensare che siamo noi programmatori a fare il mercato. Progettate buone applicazioni, innovative, tenetevi aggiornati e sicuramente i vostri sforzi verranno premiati. Troppi programmatori sono rimasti ancora legati a vecchie versioni dei linguaggi, a database ormai superati, a tecniche in disuso. È ora di aggiornarsi, di entrare in una logica molto più dinamica, dove chi si adagia sulle proprie conoscenze è perduto.

All'inizio di ogni articolo, troverete un simbolo che indicherà la presenza di codice e/o software allegato, che saranno presenti sia sul CD (nella posizione di sempre *\soft\codice* e *\soft\tools*) sia sul Web, all'indirizzo http://cdrom.ioprogrammo.it.

FILE SHARING

crea il tuo mulo da battaglia

Svelato il codice di eMule! ti spieghiamo cosa si nasconde dietro al software più usato in Internet

ARCHITETTURA La rete edonkey e il formato dei messaggi TPC/IP

CONNESSIONE Il codice per effettuare il login con i server

RICERCA I comandi per scovare musica, film, software e tutto il resto

DOWNLOAD/UPLOAD Scarica il file, salvali su disco e condividili con gli altri

pag. 16

004-005 editoriale 2-05-2007 12:17 Pagina 5

METTI WINDOWS IN SICUREZZA

Ti spieghiamo come usare il Security Configuration Manager per trovare i buchi nella tua installazione e risolverli

QUALCHE CONSIGLIO UTILE

I nostri articoli si sforzano di essere comprensibili a tutti coloro che ci seguono. Nel caso in cui abbiate difficoltà nel comprendere esattamente il senso di una spiegazione tecnica, è utile aprire il codice allegato all'articolo e seguire passo passo quanto viene spiegato tenendo d'occhio l'intero progetto. Spesso per questioni di spazio non possiamo inserire il codice nella sua interezza nel corpo dell'articolo. Ci limitiamo a inserire le parti necessarie alla stretta comprensione della tecnica.

http://forum.ioprogrammo.it

RIVISTA + LIBRO + CD-ROM in edicola

I contenuti del libro

Programmare con .NET 3.0

La novità più rilevante dell'anno in fatto di sistemi operativi è rappresentata probabilmente dall'arrivo di Windows Vista. Il nuovo OS di Microsoft sta rapidamente soppiantando Windows XP. Per noi progframmatori è tempo di aggiornare le nostre applicazioni e il miglor modo per farlo è quello di migrarle alla versione 3 del .NET framework. I vantaggi sono tanti e immediati. Prima di tutto una più moderna gestione del sistema grafico che si avvantaggia adesso di una struttura vettoriale, più internamente la possibilità di gestire con un unico strumento tutto il sottosistema di comunicazione. Infine non meno importante l'introduzione alla gestione dei flussi di lavoro. Il nuovo .NET pur essendo progettato per lavorare in coninuità con il precedente rappresenta senza dubbio un'opportunità da non perdere per rimanere al passo con i tempi. Ed è proprio in quest'ottica che si pone questo HandBook, ovvero come una guida rapida alla migrazione delle vostre applicazioni

LA GUIDA PASSO PASSO PER IMPARARE RAPIDAMENTE A SVILUPPARE CON IL NUOVO FRAMEWORK DI MICROSOFT

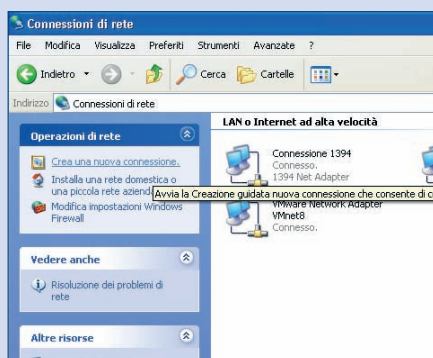
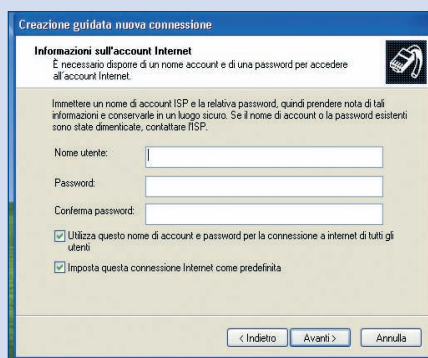
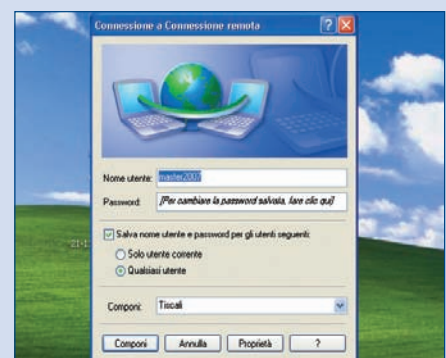
- **Introduzione a .Net Framework 3.0**
- **Windows Presentation Foundation**
- **Windows Communication Foundation**
- **Windows Workflow Foundation**

Online con Tiscali Easy

Numero unico per tutta l'Italia e nessuna registrazione. Il modo più semplice e veloce per entrare in Internet risparmiando

Sei spesso lontano da casa e hai necessità di collegarti a Internet ovunque ti trovi? Desideri salvaguardare la tua privacy navigando in modo anonimo? Non hai voglia di perdere tempo in lunghe registrazioni per creare un nuovo account? Niente paura, Tiscali ha pensato anche a te! Con Tiscali Easy arriva un sistema tutto nuovo di collegarsi a Internet. Non sarà più necessario creare un nuovo account e fornire i propri dati personali, potrai navigare collegandoti con un unico numero di telefono (7023456789) da tutta Italia e soprattutto utilizzando i dati di accesso forniti direttamente da Tiscali (UserID e Password presenti sulla scheda), quindi non collegabili in alcun modo a te. Insomma, con Tiscali Easy, otterrai in un colpo solo riservatezza dei dati, risparmio del tempo necessario alla creazione di un abbonamento e tariffe vantaggiose. I costi di connessione sono simili a quelli di un classico abbonamento gratuito: 1,90 cent. per i primi 10 minuti e 1,72 cent. per quelli successivi. Per risparmiare ulteriormente è possibile connettersi a Internet durante le ore serali o nei giorni festivi al costo di 1,09 cent. per i primi dieci minuti e 0,98 cent. per quelli successivi. L'unico requisito richiesto per poter eseguire la connessione è un modem correttamente installato. Poiché si tratta di una normale connessione analogica è possibile utilizzare i tool di accesso remoto integrati in Windows

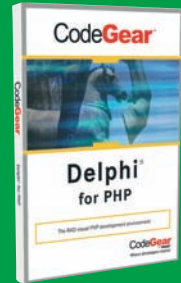
IN RETE CON TISCALI

Nessuna registrazione basta creare una nuova connessione e inserire i dati di accesso forniti da Tiscali

1 NUOVA CONNESSIONE Portiamoci nel pannello di controllo, e selezioniamo l'icona connessioni di rete. In alto a sinistra selezioniamo "nuova connessione" e prepariamoci a seguire il wizard che comparirà

2 DATI DI ACCESSO Seguiamo il Wizard fino a quando non ci vengono chiesti i dati per l'autenticazione. E' sufficiente inserire quelli presenti nella card allegata a questo numero di ioProgrammo: master2007/tiscali

3 ACCESSO A INTERNET Connettersi è semplicissimo, troveremo una nuova icona all'interno del pannello di controllo alla voce connessioni. Bastano due click ed il gioco è fatto sarete connessi ovunque vi troviate

RIVISTA + CD-ROM in edicola

DELPHI FOR PHP

Il RAD per PHP targato Borland

PHP è uno dei linguaggi più utilizzati in programmazione, soprattutto per la sua vocazione verso il web. Le nuove versioni avevano risolto qualche problema legato alla mancanza di un forte orientamento alla programmazione ad oggetti, e dalla versione 5 in poi PHP è diventato davvero un linguaggio completo e professionale. Mancava un IDE RAD a PHP. A colmare questa lacuna ci ha pensato Borland con un prodotto il cui nome Delphi per PHP la dice lunga sulle caratteristiche dell'IDE. Sostanzialmente Delphi per PHP unisce ad un editore eccezionale un fortissimo orientamento visuale e ad eventi che consente di programmare interfacce sofisticate mediante semplice trascinamento dei componenti su una form e conseguente programmazione degli eventi ad essi associati. Si tratta di un modo nuovo di programmare applicazioni PHP che potenzialmente aumenta moltissimo la produttività di ogni programmatore. Si tratta ora di cambiare radicalmente il proprio modo di sviluppare per adattarsi alla logica contenuta nel framework sottostante l'IDE.
Directory:/Delphi4php/d4php_trial.exe

Prodotti del mese

Struts 2.0.1

Il framework MVC per Java

Il pattern MVC è probabilmente il più usato da tutti gli sviluppatori in ogni linguaggio.Questo framework è il leader per quanto riguarda lo sviluppo di applicazioni Java secondo il pattern MVC. Molto comodo, stabile e affidabile, rappresenta la base di partenza per applicazioni che devono essere facilmente manutenibili. Uno standard da usare se progettate applicazioni di dimensioni generose, ma anche quando si sviluppano web application professionali.
L'impatto iniziale con il framework potrebbe non essere dei più semplici a causa della complessità dello strumento. Ma superato il gradino iniziale e compresa la logica delle action tutto diventa estremamente semplice e conseguenziale. Ne parliamo approfonditamente in questo numero di ioProgrammo

[pag.107]

Java SE Development Kit 6

Il compilatore indispensabile per programmare in JAVA

Se avete intenzione di iniziare a programmare in Java oppure siete già dei programmatori esperti avete bisogno sicuramente del compilatore e delle librerie Java indispensabili. Sotto il nome di Java SE Development Kit vanno appunto tutti gli strumenti e le librerie nonché le utility necessarie per programmare in JAVA. L'attuale versione è la 6.0, ovvero la nuovissima release densa di innovazioni e molto più legata al desktop di quanto non fossero tutte le precedenti.
Java è un linguaggio estremamente versatile la cui forza risiede nell'estrema eleganza della sua programmazione ad oggetti e la capacità di generare applicazioni più o meno per tutte le piattaforme oggi in commercio

[pag.108]

PHP 5.2.1

Il linguaggio di scripting più amato del web

Sono tre le colonne portanti di Internet: PHP, APACHE e MySQL. Certo la concorrenza è forte. Asp.NET e SQL Server avanzano con celerità, ma a tutt'oggi non si può affermare che i siti sviluppati in PHP costituiscano la stragrande maggioranza di Internet. Quali sono le ragioni del successo di cotanto linguaggio? Prima di tutto la completezza. PHP ha di base tutto quello che serve ad un buon programmatore, raramente è necessario ricorrere a librerie esterne, e quando è proprio indispensabile farlo esistono comunque una serie di repository che rendono tutto immediatamente disponibile ed in forma gratuita. Il secondo punto di forza del linguaggio sta nella sua capacità di poter essere utilizzato sia in modo procedurale che nella sua forma ad oggetti il certamente più potente e completa. Esiste un terzo di punta di forza essenziale che è quello riguardante la curva di apprendimento.

[pag.109]

php

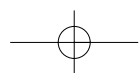
PHP 5.2.1 Release Annou

The PHP development team would
enhancement of the 5.X branch, a

Drupal 5.1

Il costruttore di Blog

Drupal è prima di tutto un sistema CMS, consente cioè la pubblicazione dei contenuti in modo semplificato, di modo che anche gli utenti meno smaliziati possano inserire dei contenuti su Internet pur non avendo conoscenze di programmazione o di html, ma Drupal è anche molto di più. Utilizzando Drupal ognuno degli utenti iscritti al sito diviene proprietario di un blog e con pochi accorgimenti potrete fare in modo che ciascuno di questi blog goda di una vita propria indipendente dal sito principale, oppure se lo desiderate potete far comunicare i vari blog in modo da creare una vera community di bloggers. Qualche parola va spesa anche sull'architettura a plugin. Sviluppare un plugin per drupal è veramente un gioco da ragazzi grazie al framework che ne è la base

[pag.109]

GLI ALLEGATI DI IOPROGRAMMO

Questo mese tre Webcast per imparare a sviluppare applicazioni a prova di hacker su ambiente Windows

La validazione dell'input

La validazione dell'input consiste nel verificare che qualsiasi parametro in ingresso nella applicazione sia conforme a quanto ci si aspetta. Per esempio un campo che accetti un indirizzo di posta elettronica deve scartare stringhe che non abbiano il formato desiderato. Oltre a migliorare la qualità dei dati, la validazione evita pericolosi attacchi come SQL Injection o Cross-Site Scripting che possono devastare il database o mettere a serio rischio la riservatezza dei dati. Nel corso del webcast vedremo come funzionano questi attacchi e come difenderci sia in applicazioni web che Windows.

Perché sviluppare pensando da subito alla sicurezza

Purtroppo ancora troppo spesso si guarda alla sicurezza come un problema secondario. In questo primo webcast sulla sicurezza vogliamo analizzare perché questo concetto sia radicalmente sbagliato e quanto sia importante pensare alla sicurezza fin dal suo progetto e prima di cominciare a scrivere codice.

Autenticazione ad autorizzazione

Autenticazione ed autorizzazione sono due processi fondamentali in sicurezza. Il primo risponde alla domanda "chi sei" mentre il secondo alla domanda "cosa puoi fare". La piena comprensione di questi meccanismi permette di risolvere problematiche molto comuni nello sviluppo di applicazioni. Per esempio i meccanismi di impersonation e delegation e i problemi e le vulnerabilità che possono derivare da un uso incorretto.

msdn WEBCAST

PERCORSI

Progettare una architettura sicura parte 1 • L'evoluzione della sicurezza nel Framework: la Code Access Security • ASP.NET 2.0 AJAX: ASP.NET AJAX Control Toolkit • 10 miti da sfatare su Windows Vista e Office System 2007 • Microsoft Solutions Framework un anno dopo: lezioni apprese, cosa ha funzionato e cosa si può migliorare nell'adozione di questa metodologia • ASP.NET 2.0 AJAX: Server Controls • Proteggere le comunicazioni in applicazioni distribuite • Tecniche di Search Engine Optimization con ASP.NET 2.0 • Autenticazione ed autorizzazione • SQL Server Advanced Express

Visita
http://www.microsoft.com/italy/msdn/risorsemsdn/security/default.mspx

FAQ

Cosa sono i Webcast MSDN?
MSDN propone agli sviluppatori una serie di eventi gratuiti online e interattivi che approfondiscono le principali tematiche relative allo sviluppo di applicazioni su tecnologia Microsoft. Questa serie di "corsi" sono noti con il nome di Webcast MSDN

Come è composto tipicamente un Webcast?
Normalmente vengono illustrate una serie di Slide commentate da un relatore. A supporto di queste presentazioni vengono inserite delle Demo in presa diretta che mostrano dal vivo come usare gli strumenti oggetto del Webcast

Come mai trovo riferimenti a chat o a strumenti che non ho disponibili nei Webcast allegati alla rivista?
La natura dei Webcast è quella di essere seguiti OnLine in tempo reale. Durante queste presentazioni in diretta vengono utilizzati strumenti molto simili a quelli della formazione a distanza. In questa ottica è possibile porre domande in presa diretta al relatore oppure partecipare a sondaggi etc. I Webcast riprodotti nel CD di ioProgrammo, pur non perdendo nessun contenuto informativo, per la natura asincrona del supporto non possono godere dell'interazione diretta con il relatore.

Come mai trovo i Webcast su ioProgrammo
Come sempre ioProgrammo cerca di fornire un servizio ai programmatori italiani. Abbiamo pensato che poter usufruire dei Webcast MSDN direttamente da CD rappresentasse un ottimo modo di formarsi comodamente a casa e nei tempi desiderati. Lo scopo tanto di ioProgrammo, quanto di Microsoft è infatti quello di supportare la comunità dei programmatori italiani con tutti gli strumenti possibili.

Su ioProgrammo troverò tutti Webcast di Microsoft?
Ne troverai sicuramente una buona parte, tuttavia per loro natura i webcast di Microsoft vengono diffusi anche OnLine e possono essere seguiti previa iscrizione. L'indirizzo per saperne di più è: www.microsoft.it/msdn/webcast. segnalo nei tuoi bookmark. Non puoi mancare.

L'iniziativa sarà ripetuta sui prossimi numeri?
Sicuramente si.

News

LAS VEGAS SBARCA IN ITALIA

Ogni anno si tiene a Las Vegas un evento denominato "Mix". Si tratta dell'evento più importante dell'anno per la presentazione delle nuove tecnologie targate Microsoft. E' durante questo evento che i maggiori esperti in informatica del mondo illustrano le tecnologie e le linee guida per il futuro. Quest'anno si terrà in Italia un evento gemello: il remix 2007. Saranno proprio gli stessi esperti del Mix a venire a presentare in Italia le nuove tecnologie. L'evento si terrà il 29 e il 30 Maggio a Milano e si avvarrà del servizio di traduzione simultanea. Il cuore dell'evento sarà dedicato allo sviluppo di applicazioni per il Web intesi sia come Designer che come programmatori. Come sempre sarà possibile fra i vari percorsi. Per chi desiderasse maggiori informazioni il link di riferimento è http://www.microsoft.it/msdn/remix.

DOMINI IT PIÙ FACILI?

Attualmente l'Italia è uno dei pochi paesi al mondo in cui la procedura di registrazione del dominio comporta l'invio di un fax e la gestione semimanuale dei dati in esso contenuti e necessari per l'approvazione. Questa procedura decisamente arcaica è diventata ormai un collo di bottiglia importante a causa del crescente numero di richieste di registrazione. Tanto che si registrano diverse petizioni a favore di un cambio radicale nella gestione delle registrazioni da parte del NIC.
Così l'ente italiano per la registrazione dei domini si è vista costretta a rispondere dichiarando che si sta effettuando una transizione importante da una gestione asincrona ad una gestione sincrona e si sta tendando anche di aumentare il personale dedicato alla registrazione dei domini. Si tratta solo di pazientare e anche l'Italia riuscirà, forse, a registrare un dominio in 15 minuti come succede per i .com e gli altri invece che in una settimana o più come avviene adesso

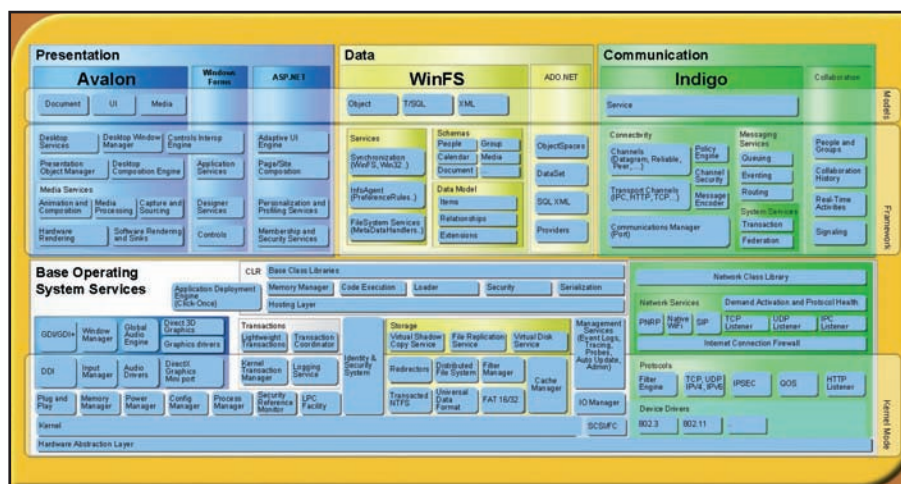
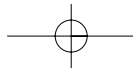
NESSUN LIMITE ALLO SVILUPPO

A dirlo è Bill Gates in persona durante il recente Intel Developer Forum di Pechino rispondendo a chi gli chiedeva, preoccupato, se ad un certo punto lo sviluppo tecnologico potesse incontrare un punto di rallentamento. Nessun rallentamento secondo Bill Gates, non esiste e non esisterà mai un momento in cui la tecnologia raggiungerà un suo limite ed in cui lo sviluppo subirà un rallentamento. Ogni limite verrà sempre costantemente abbattuto. Il proprietario di Microsoft immagina un mondo fatto di dispositivi mobili largamente interconessi, di una rete wireless che contribuirà ad abbattere gli attuali limiti della diffusione via cavo, della Televisione sempre più presente nella trasmissione via TCP/IP. Si tratta di un futuro non così lontano e quel che più conta si tratta solo di una pietra superabile alla quale si sostituiranno altri traguardi. Peccato che ad interrompere i sogni di Gates sia intervenuto un contestatore che con tanto di bandiera inneggiante al software OpenSource è salito sul palco a testimoniare la presenza del Free Software nel mondo dell'informatica. Quasi come a voler dire: il futuro è inarrestabile, l'informatica anche e l'OpenSource vuole contribuire almeno allo stesso modo in cui contribuisce Microsoft. Ovviamente il contestatore è stato ricondotto all'ordine dalle forze di polizia presenti al congresso

FLEX DIVENTA OPENSOURCE

Adobe ha recentemente annunciato di voler rilasciare l'intera piattaforma di sviluppo di applicazioni dinamiche denominata Flex con il modello OpenSource. Flex è una piattaforma che ha incontrato una larga approvazione da parte dei programmatori. Sostanzialmente consente di sviluppare intere applicazioni flash con una modalità compilativa molto diversa a quella a cui sono abituati i programmatori Flash. Con la stessa modalità si possono ottenere facilmente applicazioni che si connettono a database o fanno un largo uso di XML. Si tratta insomma di un modello che ha come risultato finale pur sempre un file SWF ma che parte da una modalità molto simile a quella utilizzata dai programmatori per scrivere il codice piuttosto che ad un ambiente grafico.
Flex sarà lisciata con modello OpenSource su tutte e tre le piattaforme più importanti: Linux, Windows, Mac OS

DISPONIBILE LA PRIMA BETA DI LONGHORN

Microsoft ha finalmente rilasciato una versione beta di quello che dovrà essere il suo sistema operativo Server del futuro. Longhorn si basa largamente sull'esperienza di vista ed aggiunge parecchi strumenti innovativi che dovrebbero aumentare stabilità e sicurezza del proprio sistema server. Le innovazioni più interessanti oltre a quelle già presente in Windows Vista riguardano la rete e la sicurezza. In particolare sarà massiccia la presenza del NAP ovvero del servizio alle cui policy tutti i partecipanti al network dovranno adeguarsi per ottenere l'accesso alla rete. Il NAP in realtà è già presente in Vista ma in Longhorn assumerà un aspetto ancora più centrale. Per quanto riguarda il ruolo di controller, il vecchio Backup Domain controller sarà affiancato dal nuovo read-only domain controller che potrà contenere una copia in sola lettura del database active directory in modo da salvaguardarne la sicurezza anche in caso di guasti hardware. Per gli amministratori di grandi reti aziendali da segnalare l'adozione di WIP come formato per la distribuzione delle immagini di installazione di Windows Longhorn. Proprio di WIP parliamo abbondantemente con un articolo minuzioso in questo stesso numero di ioProgrammo. La comunità dei sistemisti ha mostrato già un'interesse molto elevato per il nuovo sistema tanto che sono già quasi 10.000 gli iscritti al programma Beta. C'è da dire che Windows in versione Server è il prodotto più usato per quanto riguarda l'attività di Controller di Dominio e che sta scalando posizioni anche rispetto ai servizi di directory di Novell. Diversa è la situazione in ambiente Web la dove IIS pur guadagnando terreno non riesce ad avere la stessa distribuzione capillare di Apache

GOOGLE DA UNA MANO A MYSQL

Sembra che il gigante dei motori di ricerca per le proprie applicazioni abbia fatto uso molto spesso del noto serve di database MySQL. Tuttavia come è noto Google non si accontenta ma dispone di una capacità tecnica di assoluto rilievo, per cui ha sviluppato delle patch per personalizzare le proprie installazioni di MySQL. E' recente l'annuncio della disponibilità sotto licenza GPL di tali patch. In particolare si tratterebbe di alcune modifiche che coinvolgono il meccanismo di clustering e replica dei database. In ogni caso le patch sono pubblicamente disponibili all'indirizzo http://code.google.com/p/google-mysql-tools/. MySQL Ab sta prendendo in considerazione l'ipotesi di inserire queste patch nelle proprie versioni ufficiali, d'altra parte si sa che Google è un partner decisamente affidabile. Da un punto di vista prettamente filosofico è interessante annotare come la personalizzazione di strumenti OpenSource possa poi concretamente applicarsi anche a realtà dalla dimensioni decisamente generose

VIAGGIO ALL'INTERNO DEL CUORE DEL MULO

IL PEER-TO-PEER È UNO DEI FENOMENI DI MASSA DEGLI ULTIMI ANNI. LA TECNOLOGIA CHE SI NASCONDE DIETRO AL FILE SHARING È RAFFINATA QUANTO SEMPLICE. IN QUESTO ARTICOLO ANALIZZIAMO I SORGENTI DI EMULE PER CAPIRE COME POSSIAMO MODIFICARLO PER NOI...

eMule è una delle applicazioni di file sharing più popolari e utilizzate al mondo. La storia di eMule inizia nel maggio 2002, quando Hendrik Breitkreuz (conosciuto come Merkur, vedi lo splash screen di Emule), insoddisfatto del client originale per la rete eDonkey2000, inizia a lavorare al progetto.
Alla versione attuale, la 0.47, eMule consente di connettersi sia alla rete eDonkey, che alla rete Kad. Fra le caratteristiche che lo hanno portato ad essere uno dei client P2P maggiormente utilizzati, vi sono lo scambio diretto fra le sorgenti localizzate nei computer client, recupero dei download interrotti o corrotti, l'utilizzo di un sistema a crediti che incoraggia la condivisione e l'upload, la trasmissione in formato compresso zlib, per risparmiare larghezza di banda.
eMule è stato sviluppato, e continua ad esserlo, in Visual C++ (sembra strano, un'applicazione open source, programmata con un IDE Microsoft!), viene rilasciato sotto licenza GPL (GNU General Public License), ed è disponibile, tramite progetti paralleli da esso derivati, anche sotto altre piattaforme, per esempio esiste il client xMule per Linux. Sempre più lunga anche la lista di altri progetti (detti Mod) che apportano modifiche all'originale: eMule Extreme, eMule plus, aMule. A proposito, perché il nome eMule? In inglese Donkey significa "asino", quindi Merkur scelse un nome simile: Mule, cioè "mulo".

ARCHITETTURA DI EMULE

Una rete eMule è costituita ormai da centinaia di macchine che agiscono da server eMule e milioni di PC nelle case degli utenti che svolgono il ruolo di client eMule. I client devono come primo passo connettersi ad un server, per ottenere accesso alla rete eMule. La connessione al server rimarrà attiva per tutto il tempo in cui il client resterà connesso alla rete eMule.
Un server eMule esegue compiti di indicizzazione dei contenuti e non comunica mai con altri server eMule. Ogni client invece possiede una lista di server eMule, oppure può essere aggiornato con un nuovo elenco, ottenibile sia sui siti specializzati, oppure richiedendolo mediante apposito comando ad un server a cui esso possa connettersi.
Ogni client inoltre possiede un proprio elenco di file, che mette a disposizione della comunità eMule, vale a dire la lista dei file condivisi.
La connessione di un client ad un server eMule avviene utilizzando il protocollo TCP, tramite esso si ottiene l'elenco dei file disponibili e dei client sui quali questi sono presenti. Dopo aver effettuato l'accesso, mediante altre centinaia di connessioni TCP, un client comunica con i suoi pari, per effettuare l'upload o il download di file.
Tale processo è regolato mediante code di accesso ad ogni determinato file. Ogni coda è formata dai client che hanno richiesto lo stesso file, e che mano a mano che uno di essi completa il download permette l'avanzamento di un altro client. Per migliorare le prestazioni, e velocizzare il download, ogni client non effettua il download di un intero file da un singolo altro client. Ogni file è invece suddiviso in parti, ognuna delle quali può essere richiesta e scaricata da un client diverso. In tal modo possono essere sfruttate centinaia di sorgenti differenti e contemporaneamente il client può fornire le parti che ha già scaricato mettendole a disposizione di altri client.
Il protocollo eMule prevede inoltre un sistema di crediti, che avvantaggia i client che mettono a disposizione un maggiore numero di file, incoraggiando la condivisione.
Ogni server eMule mantiene un database, con le informazioni relative ai client ed ai file condivisi, ma non memorizza alcun file, agendo solo da indicizzatore.
La figura 1 rappresenta l'architettura di alto livello di una rete di server e client Emule per la condivisione di file. Un buon modo per capire il funzionamento di eMule è quello di scaricare il sorgente del client, ed eseguirlo in debug, seguendo passo passo lo scambio di messaggi. Naturalmente è necessario conoscere il linguaggio C++ e possedere Visual C++.
Nel seguito dell'articolo, implementeremo qualche spezzone di codice C#, per studiare il protocollo e verificarne le funzioni.

Fig. 1: Diagramma di una rete di server e client Emule

FORMATO DEI MESSAGGI

Ogni messaggio ha una intestazione lunga 6 byte, suddivisi in tre parti. Il primo byte indica il protocollo utilizzato, ed ha il valore 0xE3 per eDonkey, oppure 0xC5 per il protocollo eMule.
I successivi 4 byte indicano la lunghezza del messaggio in byte, escluso l'header stesso. Quindi un messaggio formato solo dall'intestazione conterrà come size il valore zero.
Infine, il terzo campo dell'intestazione, cioè l'ultimo byte, è l'identificatore univoco del messaggio stesso, un ID che indica il comando.

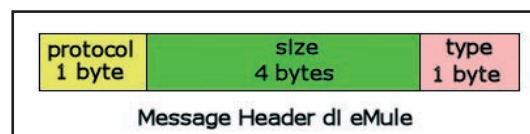



Fig. 2: L'intestazione di un messaggio Emule

Creiamo quindi una classe C# per rappresentare un'intestazione di messaggio:

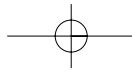
```
public class MessageHeader
{
        public Protocol.ProtocolType protocolID;
        public uint packetSize;
        public byte Command;

        public MessageHeader(byte command):base()
        {
            protocolID =
                        Protocol.ProtocolType.eDonkey;
            Command = command;
            packetSize = 6;
        }

        public void Serialize(BinaryWriter writer)
        {
            writer.Write((byte)protocolID);
            writer.Write(packetSize);
            writer.Write(Command);
        }
}
```

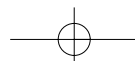
Per comodità possono essere aggiunti altri costruttori, che per brevità non riportiamo qui.
Il metodo Serialize invece ci servirà per scrivere l'intestazione su uno stream da spedire in rete.

Per appendere informazioni aggiuntive ai messaggi eMule, vengono utilizzate delle strutture chiamate Tags, e strutturate secondo una sequenza Tipo-Lunghezza-Valore (per questo vengono detti anche Tag TLV).
Esistono diversi tag differenti, documentate nel protocollo di eMule liberamente scaricabile dalla rete.
Nel seguito dell'articolo ne mostreremo ed utilizzeremo qualcuno di essi.
Ogni tag è formato da 4 campi, non tutti però vengono sempre inclusi nel corpo del messaggio:

Tipo	1 byte, indica il tipo del tag
Nome	può essere formato da una stringa di lunghezza variabile o da un intero di 1 byte;
Valore	può essere un intero di 4 byte, un numero floating di 4 byte, oppure una stringa di lunghezza variabile;
Special	è un intero di 1 byte, che identifica un tag speciale;

Tag con valori interi sono detti Integer tags, analogamente si hanno String tags e Float tags. Il tipo di un String tag è 2, di un Integer tag è 3 e di un Float tag is 4. Quando i tag vengono codificati per essere spediti in rete, viene seguito l'ordine seguente: prima il tipo, poi il nome ed infine il valore.
I nomi dati ai tag non hanno un significato particolare per il protocollo, vengono solo assegnati per un più semplice ed immediate riferimento nella descrizione dei messaggi.
Per leggere un tag implementiamo quindi una classe TagReader, in questa maniera:

```
public class TagReader
{
        public byte type;
        public byte id;
        public string name;
        public string strVal;
        public uint numVal;
        public float floatVal;
        public TagReader(BinaryReader reader)
        {
            ushort len;
            type = reader.ReadByte();
            len = reader.ReadUInt16();

            if (len == 1)
                id = reader.ReadByte();
            else
                if (len > 0) name = new
                            string(reader.ReadChars(len));

            switch(type)
            {
                case 2:
                    len = reader.ReadUInt16();
                    byte[] buf;
                    buf = reader.ReadBytes(len);
                    strVal =
                            Encoding.Default.GetString(buf);
                    break;
```

IL FILE SERVER.MET

L'elenco dei server utilizzati da eMule è contenuto in un file server.met, presente nella directory di avvio del programma. L'elenco può essere aggiornato sia fornendo l'url di un server.met presente in internet, oppure è previsto dal protocollo eMule un comando OP_SERVERLIST (0x32) che permette di effettuare una richiesta al server eMule a cui un client è connesso per richiedere una lista dei server aggiornata. Nel codice allegato all'articolo è implementato sia tale comando, si veda la classe ServerListRequest, sia l'interpretazione del formato binario di un file met.

```
            case 3:
                numVal = reader.ReadUInt32();
                break;
            case 4:
                floatVal=reader.ReadSingle();
                break;
          }
        }
      }
    }
```

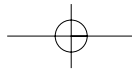
Analogamente si può implementare la classe per scrivere un tag, che non riportiamo qui per brevità ma che è possibile analizzare nel codice in allegato.

COMANDI ED OPCODE

Ogni comando eMule è identificato da uno specifico codice esadecimale, detto OpCode. Ogni comando inoltre prevede una serie di parametri da specificare. La connessione avviene inviando un comando con OpCode OP_LOGINREQUEST, pari al valore 0x01. Per evitare di specificare ogni OpCode di volta in volta, e rendere più facilmente gestibili i comandi implementeremo delle enumerazioni. Per esempio per i comandi da inviare al server una enumerazione ServerCommand con relativi opcode:

```
public enum ServerCommand : byte
{
        LoginRequest = 0x01,
        ServerMessage = 0x38,
        IDChange = 0x40,
        GetServerList = 0x14,
        OfferFiles = 0x15,
        SearchRequest = 0x16,
        ServerList = 0x32,
        SearchResult = 0x33,
        GetSources = 0x19,
        QueryMoreResults = 0x21,
        FoundSources = 0x42,
        CallBackRequest = 0x1C,
        ServerState = 0x34,
        ServerIdent = 0x41,
        CallBackRequested = 0x35,
}
```

Allo stesso modo si possono implementare enumerazioni per i comandi client, o per i valori dei campi di un tag. Per un server eMule per esempio si possono specificare ed ottenere le seguenti informazioni:

```
public enum ServerTag : byte
{
```

```
        Name = 0x01,
        Description = 0x0B,
        Ping = 0x0C,
        Priority = 0x0E,
        Fails = 0x0D,
        DynIP = 0x85,
        LastPing = 0x86,
        MaxUsers = 0x87,
}
```

CONNESSIONE AD UN SERVER EMULE

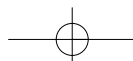
Per connettersi ad un server eMule è necessario inviare l'apposito comando di Login, con opCode LoginRequest. Innazitutto è però necessario stabilire una connessione TCP.
Per rappresentare una connessione al server creeremo una classe ServerConnection, derivata da una generica Connection, che utilizzerà i classici socket per creare una comunicazione TCP ad un indirizzo IP su una data porta.
La classe Connection possiede quindi, fra gli altri i seguenti campi:

```
public class Connection
{
        protected Socket m_socket;
        protected IPEndPoint m_EPremote;
}
```

Al metodo CreateConnection si passeranno l'indirizzo IP e la porta verso cui effettuare la connessione:

```
m_socket=new Socket(AddressFamily.InterNetwork,
SocketType.Stream,ProtocolType.Tcp);
m_socket.Blocking=false;
IPAddress ip=new IPAddress(nIP);
m_EPremote=new IPEndPoint(ip,port);
```

Mentre il metodo Connect invierà una richiesta asincrona di connessione, indicando il metodo OnConnected come metodo che si occuperà di gestire la risposta del server una volta ricevuta una risposta.

```
public void Connect()
{
try
        {
                if (!m_socket.Connected)
m_socket.BeginConnect(m_EPremote,
new AsyncCallback(OnConnected),null);
        }
        catch
        {
OnConnectionFail((byte)Protocol.ConnectionReason.C
                                annotConnect);
                }
}

protected override void OnConnected(IAsyncResult
                                        ar)
    {
        base.OnConnected(ar);
        if (m_socket.Connected)
        {
            m_Server.LastConnection
                            DateTime.Now.Ticks;
            m_Server.ResetFails();
            m_Server.IsConnected = true;
            m_Server.SendLoginRequest();
            m_Server.ConnectionReady();
        }
    }
```

All'interno del metodo OnConnected viene innanzitutto impostato l'istante dell'avvenuta connessione, ed azzerati i tentativi falliti.
Un'altra classe, Server, si occuperò di inviare e ricevere i messaggi dai server eMule. In questo primo esempio il metodo SendLoginRequest invia il messaggio di Login suddetto, costruendo un oggetto ServerLoginRequest che si occupa di riempire e formattare un buffer contenente la richiesta di login, e spedendo poi il messaggio mediante il metodo SendPacket:

```
public void SendLoginRequest()
{
        if (m_ServerConnection == null)
            return;
        MemoryStream buffer = new
                                MemoryStream();
        ServerLoginRequest hello = new
                        ServerLoginRequest(buffer);
        m_ServerConnection.SendPacket(buffer);
}

private void SendPacket(byte[] packet,int length)
{
        if
    ((!m_socket.Connected)||(m_SendPacket!=null))
        {
                return;
        }
        m_SendPacket=packet;
        m_SendBytesPending=length;
        m_TotalSent=0;
        try
        {
m_socket.BeginSend(m_SendPacket,0,m_SendBytes
                Pending,SocketFlags.None,new
            AsyncCallback(m_OnSentPacket),null);
        }
        catch
```

```
        {
OnConnectionFail((byte)Protocol.ConnectionReason.ClosedConnection);
        }
}
```

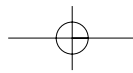
La classe ServerLoginRequest serve a formattare il messaggio di Login, costruendone l'intestazione, mediante l'utilizzo degli appropriati Tag. Il messaggio di Login, come specificato nella documentazione del protocollo eMule, deve contenere innanzitutto un intestazione e poi una serie di altre informazioni come l'id utente, versione del client, nome del client e così via. L'intestazione è costruita mediante la classe MessageHeader al cui costruttore viene passato il valore ServerCommand.LoginRequest.

```
public ServerLoginRequest(MemoryStream buffer)
{
        MessageHeader header;
        BinaryWriter writer = new BinaryWriter(buffer);
        header = new MessageHeader((byte)Protocol.ServerCommand.LoginRequest, writer);
        writer.Write(Preferences.UserHash);
        writer.Write(Preferences.ID);
        writer.Write(Preferences.TCPPort);
        uint nTags = 4;
        writer.Write(nTags);
        // user
        new TagWriter((byte)Protocol.ClientParameter.Name, Preferences.UserName, writer);
        // version
        new TagWriter((byte)Protocol.ClientParameter.Version,
Protocol.EDONKEYVERSION, writer);
        // port
        new TagWriter((byte)Protocol.ClientParameter.Port, (uint)Preferences.TCPPort, writer);
        new TagWriter((byte)Protocol.ClientParameter.EmuleVersion,
Protocol.EMULE_VERSION_COMPLEX, writer);
        header.packetSize = (uint)writer.BaseStream.Length - header.packetSize + 1;
        writer.Seek(0, SeekOrigin.Begin);
        header.Serialize(writer);
}
```

La lunghezza del messaggio varia quindi in base alla configurazione dell'utente, per esempio dal suo nickname.

Al ricevimento di un messaggio di risposta, dopo aver letto e processati i byte del messaggio.
A connessione avvenuta, il metodo ProcessIncomingPacket verifica il comando specificato nell'intestazione del messaggio, ed intraprende l'azione opportuna.

```
protected override void ProcessIncomingPacket(ref
                                    byte[] packet)
{
        if (m_Server == null) return;
        try
        {
            switch
        ((Protocol.ServerCommand)m_Header.Command)
            {
                case
            Protocol.ServerCommand.ServerMessage:
                    m_Server.SetMessage(packet);
                    break;
                case
                    Protocol.ServerCommand.IDChange:
                    uint m_ID;
                    string LowHigh_ID = "low ID";
                    bool isHighID = false;

                    BinaryReader reader = new
            BinaryReader(new MemoryStream(packet));
                    m_ID = reader.ReadUInt32();

                    if (reader.PeekChar() > 0)
                    {
                        m_Server.TCPFlags =
                                reader.ReadUInt32();
                    }
                    else
                        m_Server.TCPFlags = 0;

                    if (m_ID > 0)
                    {
                        if (m_ID >
                            Protocol.LowIDLimit)
                        {
                            LowHigh_ID = "high ID";
                            isHighID = true;
                        }
                        else
                        {
                            LowHigh_ID = "low ID";
                            isHighID = false;
                        }
ServerManager.ActiveServer.UserID = m_ID;
ServerManager.ActiveServer.IsHighID=isHighID;
                    }
                    else
                    {
OnConnectionFail((byte)Protocol.ConnectionReason.N
                                        ullID);
                    }
                    break;
                case
                Protocol.ServerCommand.SearchResult:
m_Server.ProcessSearchResults(packet);
                    break;
                case
                Protocol.ServerCommand.ServerState:
                    m_Server.UpdateState(packet);
                    break;
                case
                Protocol.ServerCommand.ServerIdent:
m_Server.ProcessDescription(packet);
                    break;
                case
                    Protocol.ServerCommand.ServerList:
m_Server.ProcessServerList(packet);
                    break;
                default:
                    break;
            }
        }
        catch (Exception e)
        {
            throw;
        }
}
```

Nel caso del messaggio di Login, verrà ricevuto come risposta un messaggio ServerMessage. Tale messaggio può avere lunghezza ed è spedito da un server in diverse occasioni, non solo dopo una richiesta di login. Un singolo server-message può contenere diverse righe di testo separate da caratteri new line, ('\r','\n' o entrambe). Messaggi che iniziano con il testo "server version", "warning", "error" e "[emDynIP: " hanno un significato per il client e possono essere trattati in maniera speciale. Gli altri messaggi possono semplicemente essere mostrati all'utente.
Nello switch del metodo ProcessIncomingPacket, esiste un case apposito per elaborare un ServerMessage, all'interno del quale viene invocato il metodo SetMessage della classe Server.
Tale metodo semplicemente interpreta i byte ricevuti, ed estrae il testo del messaggio.
Oltre all'intestazione il messaggio infatti conterrà due byte che specificano la lunghezza della stringa successiva, ed appunto una stringa di testo.

```
BinaryReader reader = new BinaryReader(new
                        MemoryStream(buffer));
ushort length = reader.ReadUInt16();
m_Message = new string(reader.ReadChars(length));
```

Il messaggio può poi essere mostrato all'utente, magari mediante una interfaccia grafica simile a quella di Mule.
La figura seguente mostra un esempio di client eMu-

le personalizzato, che sfrutta le precedenti classi e relativi metodi, subito dopo una connessione riuscita ad un server eMule.

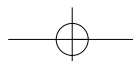
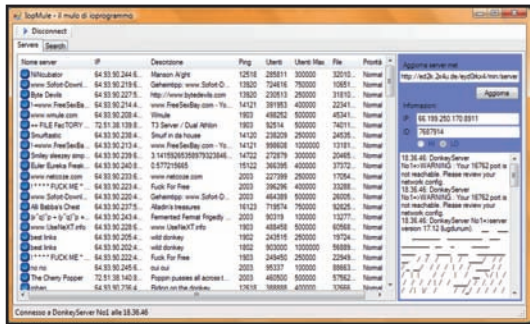



Fig. 3: Un client Emule scritto in C#

Non si esporrà nell'articolo come realizzare l'interfaccia grafica, argomento che prescinde dai nostri scopi, ricordando comunque che il progetto è incluso nel cd in allegato alla rivista.
Riprendendo il case Protocol.ServerCommand.IDChange dello switch nel precedente metodo il messaggio di IDChange viene trattato nel seguente modo:

```
BinaryReader reader = new BinaryReader(new
                        MemoryStream(packet));
m_ID = reader.ReadUInt32();
 if (reader.PeekChar() > 0)
{
m_Server.TCPFlags = reader.ReadUInt32();
}
Else m_Server.TCPFlags = 0;
```

basta leggere un intero per ottenere l'ID e poi verificare se il server ha specificato altri flag. Attualmente è possibile che il server spedisca infatti altri 4 byte, e se il valore convertito in intero è 1 indica che è supportata la compressione.

ID ALTO O BASSO?

In risposta alla richiesta di login, un server che accetta la connessione invierà anche un messaggio chiamato ID Change, che assegna un ID al client in base a determinati parametri.
Il client ID è un identificatore a 4 byte ed è valido solo per il ciclo di vita di una session di connessione client-server. I client ID possono essere distinti in alti e bassi. Un ID Basso viene in genere assegnato da un server quando questi si accorge che il client non può accettare connessioni in ingresso. Avere un ID basso diminuisce le potenzialità di utilizzo della rete eMule, diminuendo notevolemente le capacità di download, in quanto il server dovrà agire da intermediario nello scambio dei file.

L'ID viene calcolato in base all'indirizzo IP, in quanto è questo che influenza la capacità da parte di altri client di connettersi direttamente ad un'altra macchina su una data porta.
Se il server a cui ci si connette non riesce ad aprire una connessione TCP verso il client, esso assegnerà un ID basso. Ciò accade generalmente se si utilizza un firewall per bloccare determinate porte. Un ID basso si avrà anche se ci si connette attraverso NAT o proxy o se il server è troppo occupato.
Un ID viene calcolato secondo la seguente formula: se l'IP del client è X.Y.Z.W, l'ID sarà pari al valore X+2^8*Y +2^16 *Z +2^24*W. Se il valore risultante è inferiore a 16777216 (in esadecimale 0x1000000) si tratta di un ID basso.

RICERCA DEI FILE

Per avviare la ricerca di un file nella rete eMule basta utilizzare il comando Search Request, a cui il server risponderà con un comando Search Result contenente l'elenco degli eventuali risultati.
La figura 4 mostra lo scambio di messaggi per una ricerca di file.

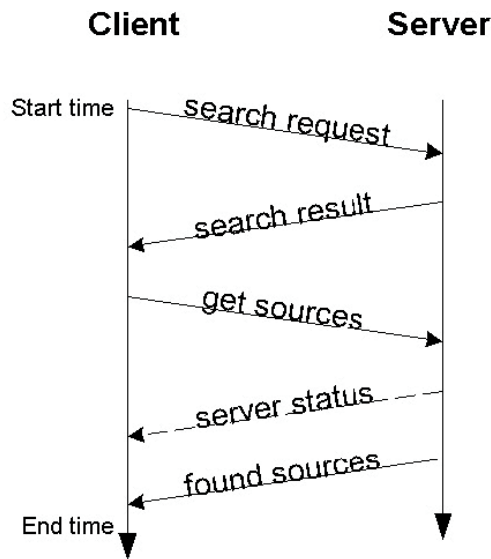
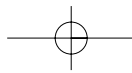
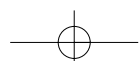



Fig. 4: Scambio di messaggi per la ricerca dei file

Nel comando di ricerca è possibile specificare i diversi parametri da utilizzare, come la dimensione massima o minima dei file, il tipo di contenuto da ricercare, l'estensione e così via.
La risposta del server è se tutto va bene un comando SearchResult, in cui viene specificato il numero n di risultati, e quindi di seguito n elementi in un determinato formato. Il codice della classe SearchResult riportato in parte qui di seguito, mostra come è possibile interpretare il messaggio dei risultati:
Il costruttore legge il numero di risultati nRes e poi

L'AUTORE

Antonio Pelleriti, ingegnere informatico, sviluppa software da più di dieci anni e si occupa di .NET sin dalla prima versione Beta. È cofondatore e chief software architect di DynamiCode©® (www.dynamicode.it), software factory che utilizza principalmente .NET per la progettazione e lo sviluppo software, oltre che content manager della .NET community www.dotnetarchitects.it. Può essere contattato per suggerimenti, critiche o chiarimenti all'indirizzo e-mail antonio.pelleriti@dotnetarchitects.it

esegue un ciclo for per ricavare i dati dei file trovati.

```
public SearchResults(MemoryStream
                        buffer,FileSearcher searcher)
{
        BinaryReader reader=new
                        BinaryReader(buffer);
        uint nRes=reader.ReadUInt32();
        for (uint i = 0; i < nRes; i++)
        {
                ExtractResult(reader,searcher);
        }
        …
}
```

Il metodo ExtractResult legge la parte seguente del messaggio interpretando i singoli tag. Per ogni risultato ricava dunque il nome, la dimensione, il numero di sorgenti, l'id del file, o altri parametri come il bitrate o l'artista se si tratta per esempio di un mp3.

```
private uint ExtractResult(BinaryReader
                        reader,FileSearcher search)
{
        TagReader tagReader;
        byte[] hash = reader.ReadBytes(16);
        uint ip = reader.ReadUInt32();
        ushort port = reader.ReadUInt16();
        uint nParams = reader.ReadUInt32();
        string fileName = "?";
        uint fileSize = 0;
        uint nSources = 1;
        string codec="";
        uint bitrate=0;
        string length="";
        bool complete=false;
        for (uint param = 0; param != nParams;
                        param++)
        {
                tagReader = new
                        TagReader(reader);
                switch(
                  (Protocol.FileTag)tagReader.id )
                {
                        case
                        Protocol.FileTag.Name:
                                fileName =
                                tagReader.strVal;
                                break;
                        case

                        Protocol.FileTag.Size:
                                fileSize =
                                tagReader.numVal;
                                break;
                        case
                        Protocol.FileTag.Sources:
                nSources = tagReader.numVal;
```

```
                                break;
                        case
                        Protocol.FileTag.Completed:
                complete = tagReader.numVal > 0;
                                break;
                        default:
                                if
(tagReader.name==Protocol.FileExtraTags.codec.ToSt
ring())
codec=tagReader.strVal;
                        else if (tagReader.name
                == Protocol.FileExtraTags.length.ToString())
                length = tagReader.strVal;
                        else if (tagReader.name
                == Protocol.FileExtraTags.bitrate.ToString())
                bitrate = tagReader.numVal;
                                break;
                }
        }
        …
        return nSources;
}
```

La figura 5 mostra la nostra applicazione subito dopo aver effettuato una ricerca del termine "acdc".

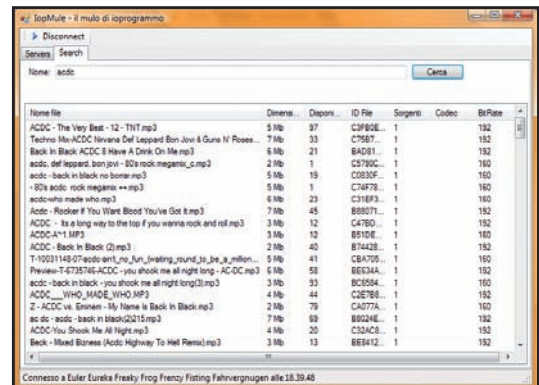
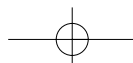
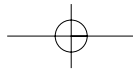



Fig. 5: Finestra con i risultati di una ricerca

CONCLUSIONI

Nell'articolo si è visto come funziona il protocollo eMule, e come sfruttare le classi messe a disposizioni da .NET per implementare dei meccanismi di comunicazione con i server della rete peer-to-peer di eMule. Al termine dell'articolo siamo in grado di implementare un nostro client per ottenere da esso il massimo delle prestazioni. La tecnologia è decisamente raffinata e studiando il protocollo si può implementare una propria rete P2P, strutturata magari appositamente per gestire le esigenze di comunicazione e di interscambio dati all'interno della propria rete aziendale. La migrazione a .NET inoltre consente performance di tutto rispetto.

Antonio Pelleriti

DOJO E JAVASCRIPT E TUTTO È SEMPLICE

IMPARIAMO AD USARE UNO DEI FRAMEWORK EMERGENTI PER AJAX. ARCHITETTURA A COMPONENTI E GESTIONE DEGLI EVENTI. AL TERMINE DELL'ARTICOLO SAREMO IN GRADO DI PROGRAMMARE ALCUNE FORM MOLTO SPECIALI. VEDIAMO COME...

Dojo è uno dei framework Javascript più completi e versatili, ed è liberamente scaricabile dal sito http://dojotoolkit.org .
Per mezzo di tale framework possiamo aggiungere alla nostra applicazione, in modo davvero semplice e intuitivo, innumerevoli funzionalità interattive basate su sofisticati componenti DHTML, drag & drop, animazioni, richieste Ajax.
Similmente ad altri framework particolarmente ambiziosi ed "impegnati", anche Dojo si basa "pesantemente" sull'approccio alla programmazione orientata ai componenti, e questo aspetto, com'è facile intuire, semplifica molto la vita agli sviluppatori che se ne avvalgono per realizzare complesse web-application. I componenti DHTML forniti da Dojo ("widget") sono inseribili nella pagina utilizzando un semplice linguaggio di markup, molto intuitivo e rapido da apprendere, che non obbliga lo sviluppatore a conoscere il funzionamento dell'architettura sottostante. L'aspetto visuale dei vari widget è poi liberamente modificabile agendo sui template html e sui file CSS, forniti a corredo dei componenti stessi.
L'efficiente struttura a "package" (librerie) su cui è basato il framework, infine, grazie alla modularità che la caratterizza, permette poi allo sviluppatore di includere nell'applicazione soltanto le classi veramente necessarie, a tutto vantaggio quindi dell'organizzazione e della pulizia del codice.
"Fin qui, niente di nuovo", starà sicuramente pensando qualche lettore che in fatto di framework ha già una discreta familiarità. A dire il vero, infatti, sono molte le librerie che presentano caratteristiche e vantaggi simili a quelli appena elencati. Ma quali sono, allora, i reali vantaggi, rispetto alla maggior parte dei framework, che possono incoraggiare il lettore a puntare su una soluzione complessa ed avanzata come Dojo ?
Ebbene, intanto bisogna ricordare che Dojo garantisce la portabilità dei propri componenti non soltanto in HTML, ma anche in altri formati, come SVG, aspetto questo di non scarsa rilevanza in uno scenario, come quello attuale, sempre più orientato a far conciliare e integrare tra di loro le più differenti tecnologie Web.
Da rimarcare, peraltro, la grande completezza e versatilità del framework, che presenta una straordinaria varietà di componenti, alcuni dei quali davvero molto interessanti. Solo per citarne due dei più rappresentativi:

- un completo editor WYSIWYG di testi ("dojo.widget.Editor"), facile da includere nella pagina, che consente di lavorare con il browser come se si trattasse di un vero e proprio programma di editing;
- un potente e versatile motore per la generazione di grafici in formato SVG ("dojo.widget.Chart").

Vediamo ora di analizzare un po' più nel dettaglio il funzionamento del framework Dojo, partendo naturalmente dalle principali basi teoriche.

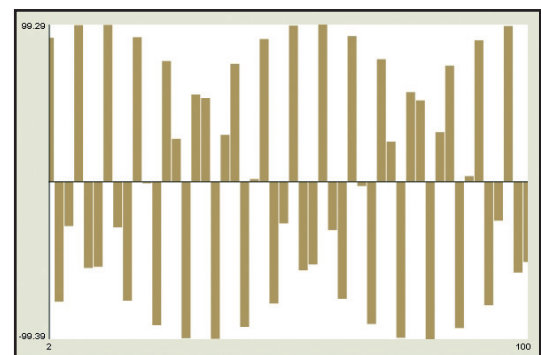



Fig. 2: Ecco un esempio di grafico SVG elaborato da Dojo:

FUNZIONI DI UTILITÀ DI DOJO

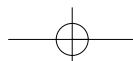
Come ogni buon framework che si rispetti, Dojo offre una vasta gamma di funzioni di utilità,

REQUISITI

Conoscenze richieste
elementi di Javascript, Php, Ajax

Software
un browser recente, un web-server, Dojo framework

Impegno

Tempo di realizzazione

destinate a molteplici scopi.
Prima di poter usare le utility di Dojo, però, dobbiamo linkare il file che ne include l'intera logica di funzionamento:

```
<script type="text/javascript"
                         src="dojo.js"></script>
```

Vediamo ora le principali librerie generiche che ci mette a disposizione Dojo.
Innanzi tutto, va citata la dojo.dom, che come si può facilmente intuire fornisce utili scorciatoie e semplificazioni per l'accesso al Document Object Model.
Se però siamo interessati più che altro alla manipolazione di tag HTML, dobbiamo allora utilizzare la dojo.html, della quale citiamo anche alcune funzioni particolarmente significative:

- **getScroll():**

restituisce un oggetto che ha le proprietà top e left valorizzate sulla base dello scrollTop e dello scrollLeft del documento HTML;

- **getElementsByClass(className, parent, nodeType):**

restituisce tutti i tag HTML (discendenti dal nodo opzionale parent, rappresentato di default dal root del documento) che hanno il className e il tipo di nodo specificati;

- **addClass(node, className):**

aggiunge una nuova classe className ad un nodo, sovrascrivendo gli stili di altre classi preesistenti (per effettuare la rimozione delle classi specificate dobbiamo invece usare removeClass).
Dojo ci permette poi di lavorare agevolmente anche con gli stili CSS, tramite le funzioni apposite contenute nella libreria dojo.style; per citarne una:

- **getAbsolutePosition (node, includeScroll):**

restituisce la posizione assoluta dell'elemento node all'interno del body, come somma degli offset dei vari elementi contenitori (inclusa, eventualmente, la barra di scorrimento, la cui presenza dipende dal valore del parametro booleano includeScroll).
Per includere una classe (o un intero namespace) nella pagina, è sufficiente (dopo aver linkato il file dojo.js) usare l' istruzione dojo.require. Es.

```
dojo.require("dojo.dom");
dojo.require("dojo.event.*");
```

Insomma, è evidente che l'utilizzo di queste funzioni "prefabbricate" agevola notevolmente l'attività dello sviluppatore, che non deve più preoccuparsi delle incompatibilità tra i vari browser.
Vediamo ora come lavorare con gli eventi Dojo all'interno della nostra pagina HTML. Intanto, è bene sapere che per agganciare un event handler ad un elemento html possiamo utilizzare semplicemente il seguente metodo:

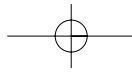
```
dojo.event.connect(document.getElementById("node
                    1"), "onclick", "handleOnClick");
```

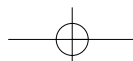
La riga di codice sopra indicata associa all'evento onclick dell'elemento di id "node1" la funzione handleOnClick.
Se invece vogliamo associare all'evento una funzione anonima, che quindi intendiamo definire sul momento, possiamo agire nel seguente modo:

```
dojo.event.connect(document.getElementById("node
                    1"), "onclick", function(evt) {
 // evt rappresenta un riferimento all'evento valido
                               per tutti i browser
});
```

Senza addentrarci troppo nei meandri della programmazione ad oggetti in Javascript, è bene comunque precisare che la funzione dojo.event.connect permette anche di associare una molteplicità di event handler a cascata (ad esempio, una serie di metodi statici appartenenti ad un oggetto che chiamerò obj) al verificarsi di un determinato evento (o anche solo su chiamata di una semplice funzione), attraverso un codice simile al seguente:

```
dojo.event.connect(obj, "funzione1", obj,
                                  "funzione2");
```

Il metodo funzione2, in parole semplici, è una funzione callback che viene richiamata automaticamente al termine dell'esecuzione del metodo funzione1 (pur non essendo, quest'ultimo, un evento a tutti gli effetti).
Come si può vedere, quindi, invocando una

ANIMAZIONI CON DOJO

**Naturalmente Dojo integra anche funzionalità per la gestione delle animazioni. Le possibilità offerte dal framework annoverano, tra l'altro, anche librerie legate alla matematica (come dojo.math), che consentono la realizzazione di animazioni con figure geometriche complesse.
Tuttavia, se l'obiettivo è inserire dei semplici effetti animati, allora si può tranquillamente ricorrere alla classe dojo.graphics.htmlEffects, che integra diversi metodi per l'implementazione delle più comuni tecniche di animazione.**

semplice API, possiamo evitare di dover gestire tutte le differenti implementazioni degli eventi tipiche dei vari browser, concentrandoci invece solo sui nostri reali obiettivi.
A questo punto, finalmente, avendo dato uno sguardo rapido ma efficace a tutte le principali funzionalità offerte da Dojo, abbiamo gli elementi per introdurre l'argomento clou della nostra trattazione sul framework: i widget.

COSTRUIAMO UN EDITOR PER IL WEB

Quante volte, dovendo realizzare un form per l'invìo di dati, abbiamo pensato all'opportunità di inserire nella pagina un editor in grado di consentire all'utente una qualche formattazione dei testi inseriti. Ma ben presto l'entusiasmo si spegneva di fronte alla difficoltà di trovare componenti compatibili con i principali browser, e in grado di assicurare stabilità e sicurezza nelle più svariate situazioni.
Ebbene, l'editor fornito da Dojo è nato proprio per rispondere a queste particolari esigenze. Esso, infatti, non soltanto consente di lavorare direttamente sull'output finale della formattazione (lasciando quindi il "lavoro sporco" di generazione del codice html al framework), ma è in grado anche di comportarsi in modo stabile e sicuro in qualsiasi situazione, senza far correre il rischio di perdere dati preziosi.
Per poter includere l'editor, innanzi tutto, occorre inserire nella head della pagina il seguente codice:

```
<script src="dojo.js"
                    type="text/javascript"></script>
<script type="text/javascript">
   dojo.require("dojo.widget.Editor");
</script>
```

CRONOLOGIA E SEGNALIBRI CON AJAX

Il grande inconveniente dell'utilizzo di Ajax è legato al fatto che, quando interviene una modifica della pagina da remoto, l'URL non cambia, e quindi i tasti Back e Forward non funzionano più come nei siti web tradizionali. Dojo consente di intercettare la pressione dei due pulsanti, mediante le funzioni backButton e forwardButton del metodo dojo.io.bind, che l'utente può costruire a proprio piacimento. In questo modo è possibile implementare una forma personalizzata di cronologia che tenga conto anche delle azioni Ajax intervenute all'interno della pagina. Per preservare la funzionalità dei segnalibri, infine, è poi possibile specificare un parametro all'interno del metodo dojo.io.bind (changeURL: true) che ad ogni chiamata remota aggiunge all'URL della pagina il simbolo # seguito da un timestamp: questo assicurerà, evidentemente, URL diversi ad ogni richiesta, rendendo possibile l'uso dei bookmark.

Un editor Dojo è costituito da due widget: Rich Text (che contiene il testo da formattare) e Toolbar (che rappresenta la barra con i vari pulsanti). Inserire un editor Dojo nella pagina è decisamente molto semplice: è possibile, infatti, assegnare la classe dojo-Editor ad un DIV, per fare in modo che questo si trasformi in un editor in piena regola:

```
<div class="dojo-Editor"> Questo &egrave; il
                              contenuto </div>
```

Oppure si possono inserire degli attributi proprietari nel tag del DIV, come avviene ad esempio nel seguente caso:

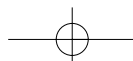
```
<div id="editable" dojoType="Editor"
                          widgetId="editdiv"
 items="formatblock;|;insertunorderedlist;insertorde
 redlist;|;bold;italic;underline;strikethrough;|;create
                                          Link;">
```

L'attributo dojoType, in particolare, specifica il tipo di widget da istanziare, mentre widgetId rappresenta l'identificativo di quel widget all'interno della pagina.
L'attributo items, infine, contiene l'indicazione dei vari pulsanti presenti all'interno della toolbar, separati tra di loro da segni di interpunzione (;) e suddivisi in raggruppamenti logici attraverso il simbolo pipe (|).

Fig. 2: Ecco un esempio di editor Dojo in azione

Ma come fare poi per recuperare il contenuto del nostro editor ? Avevamo accennato, infatti, che l'editor fa parte di un form contenente vari campi da inviare al server al termine della compilazione.
Niente paura: finora abbiamo usato il tag DIV, per generare l'editor, ma nulla ci vieta di usare altri elementi HTML, ad esempio una textarea:

```
<textarea dojoType="Editor" name="textCont">
   ecco la nostra textarea trasformata in un editor
</textarea>
```

Evidentemente, il valore da inviare al server su submit del form sarà costituito dal codice html generato da Dojo, e non quindi dal semplice testo inserito all'interno della textarea.

RICHIESTE AJAX CON DOJO

A dire il vero, l'utilizzo dei moduli non è l'unico sistema per recuperare il contenuto dell'editor. L'invio del contenuto HTML generato dal controllo (che si può interrogare, ad esempio, attraverso il metodo getEditorContent()), può avvenire anche attraverso una chiamata asincrona al server. A tale scopo possiamo utilizzare il metodo dojo.io.bind, presente all'interno del namespace dojo.io (che contiene le funzionalità di I/O).
Lo strato di remoting fornito da Dojo incapsula già tutta la logica di gestione delle varie implementazioni di Ajax nei differenti browser, mettendoci a disposizione API ad alto livello estremamente stabili e semplici da utilizzare.
Qualora, poi, non sìa possibile istanziare l'oggetto giusto per l'impiego di Ajax, allora il framework si avvale automaticamente del vecchio sistema basato su iframe, in modo da consentire in ogni caso il corretto funzionamento dell'applicazione.
Ecco un esempio di invio del contenuto di un form tramite Dojo:

```
dojo.io.bind({
        url: "http://...",
        load: function(type, obj) { /* questa è la
              risposta in caso di esito positivo */ },
        error: function(type, error) { /* questa è
                   la risposta in caso di errore */ },
        formNode:
            document.getElementById('nameForm')
}
```

In particolare, il parametro *type* identifica l'esito della risposta (se è positivo o meno), mentre il secondo parametro rappresenta il responseText dell'oggetto XmlHttpRequest (nel caso della Load) o una chiave per la gestione degli errori (nel caso di Error).
Esistono poi anche altre opzioni, tramite le quali possiamo specificare, ad esempio, il metodo della chiamata (post o get), l'invio di parametri di richiesta sotto forma di hash (content), la modalità di invio dei dati (sincrona o asincrona), l'utilizzo di una eventuale cache con cui gestire gli oggetti XmlHttpRequest, ecc...

DOJO, IL FUTURO DEL WEB 2.0

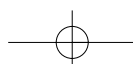
Dire che Dojo rappresenta il futuro del Web 2.0 può sembrare azzardato, ma è comunque fuori discussione che esso presenta delle funzionalità davvero innovative, ancora pressocché introvabili, a tutt'oggi, nell'ambito della comunità open source. Mi riferisco, evidentemente, a Cometd, una libreria che consente di attuare un approccio alle comunicazioni asincrone decisamente ottimizzato (almeno per quello che riguarda le applicazioni "collaborative", come ad esempio una chat, dove cioè il client deve interrogare continuamente il server per aggiornarsi).
La grande innovazione apportata da Ajax (se mai si può definire una vera e propria innovazione) si limita, in effetti, a rendere asincrona l'interazione con il server, ma tale approccio resta di fatto sempre e comunque basato sulla logica client-server del protocollo HTTP.
Una volta effettuato lo scambio di comunicazioni tra client e server, infatti, la connessione muore. L'approccio basato su Cometd si basa, invece, su una connessione persistente, e sulla possibilità, da parte del server, di inviare una comunicazione al client di propria spontanea iniziativa, al verificarsi di determinati eventi, evitando quindi continue e inutili richieste di auto-aggiornamento.
Il grosso inconveniente di questo approccio alla programmazione web-oriented, che ne limita ancora per il momento la diffusione, è la necessità di dover disporre, sul server, di un componente in grado di gestire le code di messaggi Cometd (come ad esempio la soluzione open source Jetty, il cui sito web di riferimento è http://www.mortbay.com/ software/Jetty.html); tuttavia l'idea resta indubbiamente di notevole interesse, vista la grande novità che rappresenta rispetto alle soluzioni Ajax-based attuali, ed è molto probabile che la prossima rivoluzione del Web 2.0 volga proprio in tale direzione.
Insomma, è evidente che, pur essendo ancora in evoluzione, Dojo ha già tutte le carte in regola per stupire: motivo in più, questo, per seguirne attentamente gli sviluppi futuri, nella speranza che la grande portata innovatrice di questo framework possa dare un contributo valido e determinante all'intero mondo del Web 2.0.

Enrico Viale

L'AUTORE

Enrico Viale è specializzato nello sviluppo di applicazioni sia web-oriented sia desktop. Chi desidera contattarlo per chiarimenti riguardo all'articolo, o per qualsiasi altro motivo, può farlo all'indirizzo enrico.viale@gmail.com.

COS'È SVG ?

SVG è l'acronimo di Scalable Vector Graphics, ed è una tecnologia raccomandata come standard W3C nel 2001 per la generazione di grafica vettoriale. Tale tecnologia consente di realizzare immagini vettoriali, in modo simile a Macromedia Flash, ed è basata totalmente su XML. SVG è supportato nativamente da tutti i browser moderni, tranne in qualche caso dove è possibile, comunque, installare componenti esterni che ne garantiscono comunque le funzionalità.

EFFETTI GRAFICI AL VOLO PER IL WEB

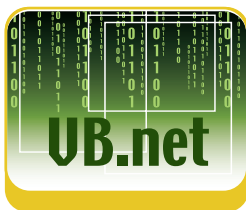
IMPAREREMO COME CAMBIARE DINAMICAMENTE LO SFONDO DELLE NOSTRE PAGINE. INOLTRE FAREMO IN MODO DI UTILIZZARE UN EFFETTO GRADIENTE E DIMINUIRE IL PESO DELLE IMMAGINI SFRUTTANDO IN MODO PARTICOLARE I FOGLI DI STILE

L'introduzione dei fogli di stile CSS ha portato un notevole progresso nel design dei siti Web moderni.
Una grande quantità di effetti grafici che prima richiedevano l'utilizzo di immagini, adesso possono essere ottenuti con un semplice CSS.
Uno degli effetti più "in voga" tra i web designer è quello dell'utilizzo dei gradienti di colore, basti vedere alcuni dei siti più famosi come quello di Yahoo (**figura 1**).

Fig. 1: Utilizzo dei gradienti nel design del sito di Yahoo

Purtroppo alcuni effetti grafici, e tra questi proprio i gradienti di colore, non sono ottenibili con il solo uso dei fogli di stile.

IMPOSTARE I GRADIENTI CON I CSS

L'impostazione dei gradienti, come sfondo ad alcuni elementi HTML, richiede normalmente, oltre ai CSS, l'utilizzo di file grafici (tipicamente JPG) di supporto.
Prendiamo una semplice tabella HTML:

```
<table border="1">
  <tr>
    <th>Nome</th>
    <th>Cognome</th>
    <th>Indirizzo</th>
  </tr>
  <tr>
    <td>Mario</td>
    <td>Bianchi</td>
    <td>Via Verga,3 Milano</td>
  </tr>
…
</table>
```

Poniamo di volere applicare un gradiente di colore alla testata ed uno alle righe. Procederemmo creando, in Photoshop o altro editor di immagini, due file grafici con i gradienti desiderati salvandoli su disco con i nomi, ad esempio, *gradiente1.jpg* e *gradiente2.jpg* (**figura 2**).

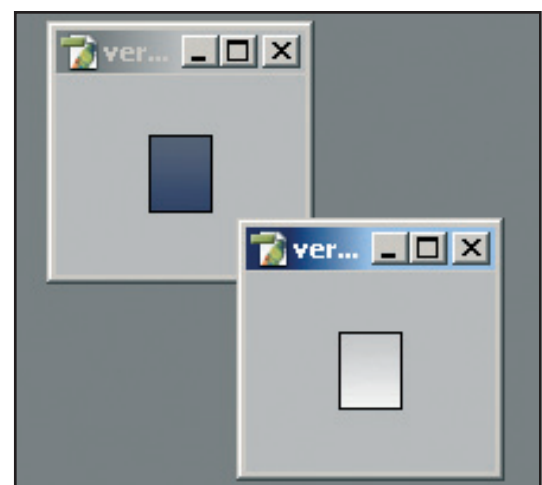
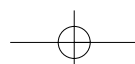



Fig. 2: Creazione dei gradienti in Photoshop

a questo punto, nel foglio di stile è sufficiente impostare i file come background degli elementi in questione definendo una classe che chiameremo myTable:

```
.myTable {
        margin-top:10px;
        border: 1px #666 solid;
        border-bottom: none;
    }

    .myTable TH {
        background-image:
            url('images/gradiente1.jpg');
        background-position: left;
        background-repeat: repeat-x;
        background-color: #999;
    }
    .myTable TD {
        background-image:
            url('images/gradiente2.jpg');
        background-position: left;
        background-repeat: repeat-x;
        background-color: #FFF;
    }
```

cioè, in pratica, utilizziamo la proprietà background-image per definire quale sarà lo sfondo dell'elemento.
Per collegare quindi la classe CSS alla nostra tabella sarà sufficiente utilizzare l'attributo class nel codice HTML:

```
<table class="myTable">
```

A questo punto potremo ammirare la nostra tabella "prima" e "dopo" la cura come mostrato in **figura 3**.

Fig. 3: Tabella formattata con i CSS

SOLUZIONI PROPRIETARIE

Come abbiamo visto la tecnica di base per la realizzazione dei gradienti di sfondo con i CSS non è affatto complessa, tuttavia presenta anche qualche piccolo inconveniente:

- Per cambiare i colori di un gradiente dovremmo editare nuovamente l'immagine
- Molte volte vorremmo utilizzare la stessa sfumatura di colore per elementi di dimensione diversa e quindi siamo costretti a realizzare più immagini nelle varie dimensioni

Poiché il gradiente è un effetto grafico molto "smart" per i siti, Microsoft ha aggiunto questo effetto nei *Visual filters*.
I *Visual filters* sono, in pratica, delle estensioni ai CSS standard mediante l'attributo filter, riconosciuto e gestito da Internet Explorer fin dalla versione 4. La sintassi CSS di questo attributo è un po' particolare :

```
filter:
progid:DXImageTransform.Microsoft.gradient(startCo
        lorstr=#0000FF, endColorstr=#FFFF00,
                gradientType=0);
```

Le proprietà fondamentali (espresse all'interno delle parentesi) sono :

- *startColorstr* – colore di inizio in formato esadecimale
- *endColorstr* – colore finale in formato esadecimale
- *gradientType* – tipo di gradiente 0 (verticale) e 1 (orizzontale), il valore di default è 0.

Tornando all'esempio fatto in precedenza, la classe CSS per formattare la nostra tabella utilizzando l'attributo *filter* si presenterebbe così :

```
.myTableIE {
        margin-top:10px;
        border: 1px #666 solid;
        border-bottom: none;
}

.myTableIE TH {
filter:progid:DXImageTransform.Microsoft.gradient(st
        artColorstr=#5D6075,
        endColorstr=#3D4167,gradientType=0);
}

.myTableIE TD {
filter:progid:DXImageTransform.Microsoft.gradient(st
        artColorstr=#CFCFCF,
        endColorstr=#F7F7F7,gradientType=0);
}
```

In **figura 4** vediamo il risultato : in IE utilizzando *filter* otteniamo lo stesso risultato rispetto all'uso di immagini esterne, ma Firefox non riconosce la formattazione.

RIFERIMENTI PER IL BACKGROUND CSS

Un ottimo riferimento sui background CSS (e i CSS in generale) è reperibile su http://www.w3schools.com/css

Fig. 4: Comparazione dei vari metodi in IE e Firefox

In effetti il problema è proprio questo: la tecnica avrebbe tutte le carte in regola per rappresentare la soluzione del problema (elimina la necessità di immagini esterne ed è facilmente modificabile editando solo il CSS), tuttavia non possiamo dimenticarci della crescente quota di utenti che usano Firefox o di chi non usa Windows.

VISUAL FILTERS CSS

Chi fosse interessato ad approfondire i Visual filters introdotti da Microsoft può trovare i riferimenti su http://msdn2.microsoft.com/en-us/library/ms532847.aspx

LA GENERAZIONE DINAMICA DI GRADIENTI

La tecnica proposta da Microsoft in IE ci dà comunque degli spunti interessanti, soprattutto nel senso di obiettivi da raggiungere:

1. generazione dinamica e non statica
2. modifica del solo CSS e non di immagini
3. manutenibilità

a questi obiettivi però aggiungiamo anche quello della portabilità in tutti i browser (che naturalmente supportino i CSS) e le piattaforme.
La soluzione è rappresentata dal cd. “lato server” cioè dalla generazione dinamica di immagini sul server anziché sul client (come avviene nel caso dei *Visual filters* di Microsoft).
Piattaforme come ASP.NET sono prevalentemente utilizzate per realizzare dinamicamente output di tipo HTML e spesso dimentichiamo che possono essere agevolmente sfruttate per produrre *Streams* di qualunque tipo, anche immagini.
Quello che vogliamo ottenere quindi è un programma lato server, rispondente ad una URL, che restituisce un’immagine con il gradiente desiderato costruito dinamicamente in base ad alcuni parametri passati.

In tal modo potremmo usare i CSS nel modo normale, ad esempio:

```
background-image:
 url('images/gradient.aspx?startColor=#CCCCCC&endColor=#FFFFFF');
```

con la sola differenza che variando i parametri cambierebbe il gradiente.
Mettiamo quindi mano al codice e creiamo il nostro “generatore di gradienti” in .NET.
In Visual Studio impostiamo un nuovo progetto Web e vi inseriamo subito un nuovo elemento.
Al contrario di quello che possiamo pensare, non sarà una pagina ASP.NET ma un file di tipo “Gestore generico” (**figura 5**).

Fig. 5: Nuovo file di tipo “Gestore generico”

Tale tipo di file è riconoscibile dall’estensione .ASHX anziché .ASPX ed è detto HttpHandler. Utilizziamo un *HttpHandler* invece di una pagina ASP.NET perché, nel nostro caso, non abbiamo bisogno di interfaccia HTML e controlli vari: è sufficiente gestire la *Pipeline* Request/Response.
Un *HttpHandler* è una classe di struttura molto semplice, che implementa l’interfaccia IHttpHandler, questo è lo schema predefinito creato da Visual Studio:

```
<%@ WebHandler Language="VB" Class="Handler" %>
Imports System
Imports System.Web
Public Class Handler : Implements IHttpHandler

    Public Sub ProcessRequest(ByVal context As HttpContext) Implements IHttpHandler.ProcessRequest
        context.Response.ContentType = "text/plain"
        context.Response.Write("Hello World")
    End Sub
    Public ReadOnly Property IsReusable() As Boolean Implements IHttpHandler.IsReusable
        Get
```

```
            Return False
        End Get
    End Property
End Class
```

La proprietà IsReusable definisce se la classe può essere utilizzata da un'altra richiesta, nel nostro caso è false perché il gradiente generato dovrà essere diverso a seconda dei parametri passati.
Il tutto si gioca poi a livello del metodo ProcessRequest che regola il flusso di risposta al client.
Nel nostro caso definiremo prima di tutto quali sono i parametri richiesti:

```
    Dim GradientType As String ‘tipo gradiente
                                    orizzontale o verticale
    Dim StartColor As Color ‘ colore di inizio
    Dim EndColor As Color ‘ colore finale
    Dim GradientLength As Integer ‘la larghezza o
        l’altezza del rettangolo contenente il gradiente
```

Sempre a livello di classe saranno definite poi altre variabili:

```
    Dim GradientRect As Rectangle ‘il rettangolo su
                                    cui disegnare il gradiente
    Dim Mode As Drawing2D.LinearGradientMode ‘il
        tipo di gradiente espresso come valore enum
```

I parametri saranno ricavati dalla QueryString. Qui abbiamo una particolarità: nella url il carattere # non si può usare così come nella QueryString perché sarebbe interpretato come riferimento ad un Bookmark, quindi i colori in formato esadecimale dovrebbero essere espressi con i caratteri di escape, ad esempio come %23FFFFFF invece che # FFFFFF, preferiamo allora omettere del tutto il cancelletto e realizzare una funzione che, al bisogno, lo aggiunge:

```
    Private Function TryParseColor(ByVal s As
                                    String) As Color
        Dim result As Color = Color.Black
        Try
            result = ColorTranslator.FromHtml(s)
        Catch ex As Exception
            If Not s.StartsWith("#") Then
                result = TryParseColor("#" & s)
            End If
        End Try
        Return result
    End Function
```

in questo modo, anche se nella nostra URL i cancelletti non ci sono, in fase di valutazione della stringa verranno aggiunti.
Tutti i parametri vengono recuperati nel metodo *ProcessRequest*, all’interno del quale viene definito anche il rettangolo su cui disegnare.
Il disegno vero e proprio del gradiente viene demandato a una funzione:

```
    Private Function GetGradient() As Bitmap
        Dim bmp As New Bitmap(GradientRect.Width,
                                GradientRect.Height)
        Dim gBrush As New
        Drawing2D.LinearGradientBrush(GradientRect,
                            StartColor, EndColor, Mode)

        Dim g As Graphics =
                            Graphics.FromImage(bmp)
        g.FillRectangle(gBrush, GradientRect)
        Return bmp
    End Function
```

Naturalmente rimandiamo agli esempi allegati per un esame più completo del codice, ci basti dire qui che, una volta costruito il gradiente, nel metodo *ProcessRequest*, definiremo la scrittura dell’immagine nell’output:

```
        Dim bmp As Bitmap = GetGradient()
        response.ContentType = "image/jpeg"
        bmp.Save(response.OutputStream,
                            Imaging.ImageFormat.Jpeg)
```

UTILIZZO DEL GENERATORE

I nomi dei parametri che abbiamo definito nel codice sono:

- *start* – colore di partenza del gradiente in formato esadecimale (#FFFFFF) o named colours (white)
- *end* – colore di arrivo del gradiente in formato esadecimale o named colours
- *direction* – direzione del gradiente (*horizontal* e *vertical*)
- *size* – se il gradiente è orizzontale rappresenta la larghezza del rettangolo altrimenti l’altezza, l’altro lato invece sarà sempre di 10px perché l’immagine può essere ripetuta attraverso i CSS.

Quindi la URL per richiamare il nostro generatore sarà qualcosa di simile a:

```
gradient.ashx?direction=horizontal&size=300&start
                                    =00FF00&end=red
```

e, se carichiamo la URL nel browser, avremo come

NOTA

USO DEL CODICE ALLEGATO

Per utilizzare il codice allegato al CD è necessario salvare la directory con gli esempi nel disco fisso e configurarla come directory virtuale di IIS nella macchina di destinazione

ANCHE IN PHP

La tecnica illustrata in questo articolo può essere utilizzata anche con altri linguaggi lato server che dispongano di funzioni grafiche. In particolare in PHP esiste una piccola classe che genera gradienti scaricabile da http://frenchfragfactory.net/ozh/my-projects/images-php-gd-gradient-fill.

risultato l'immagine che appare in **figura 6**.

Fig. 6: il generatore in "azione"

a maggior conferma del fatto che si tratta di una vera e propria immagine, del tutto simile a una JPG statica possiamo usarla nel codice HTML proprio come faremmo con qualsiasi altra immagine:

```
<img
src="gradient.ashx?direction=vertical&size=300&start=66ff00&end=ccff66">
```

USO CON I CSS

Passiamo quindi al vero e proprio scopo del nostro generatore di gradienti: quello di essere usato in combinazione con i CSS.
Riprendiamo quindi il nostro primo esempio, i gradienti utilizzati come sfondo di una tabella. Ridefiniamo quindi la classe CSS della tabella utilizzando la URL del nostro generatore al posto delle immagini statiche:

```
.myTable {
margin-top:10px;
border: 1px #666 solid;
border-bottom: none;
}
.myTable TH {
background-image:
  url('gradient.ashx?direction=vertical&size=30&start=5D6075&end=3D4167');
background-position: left;
background-repeat: repeat-x;
}

.myTable TD {
background-image:
  url('gradient.ashx?direction=vertical&size=30&start=CFCFCF&end=F7F7F7');
background-position: left;
```

```
background-repeat: repeat-x;
}
```

Il risultato che otterremo (**Figura 7**) sarà del tutto simile al precedente con il vantaggio però che per cambiare i colori basterà cambiare i valori dei parametri anziché sostituire le immagini. Insomma un metodo del tutto simile al *filter* di IE, ma compatibile con tutti i browser!

Fig. 7: il generatore in combinazione con i CSS

PERFEZIONAMENTI

Il fatto di avere a disposizione una piattaforma server come .NET ci porta a risolvere con successo un altro problema, quello della collocazione del file.
L'indirizzo dell'immagine impostata come background nel CSS è relativo alla collocazione del CSS, quindi se l'immagine ha un percorso tipo */immagini/bg.jpg* e il foglio di stile uno come */css/style.css* in quest'ultimo il background andrà impostato come:

```
background-image: url('../immagini/bg.jpg ');
```

se però sposto il foglio di stile in un'altra posizione, ad esempio in /css/altrapath/style.css, il riferimento dovrà essere :

```
background-image: url('../../immagini/bg.jpg ');
```

Questo ovviamente riduce la "portabilità" del foglio di stile perché costringe a cambiare le path; senza parlare dei casi in cui si utilizza una classe CSS direttamente nel documento HTML con il tag <*STYLE*> o con l'attributo *style* di un elemento.
Il problema non è risolto con il generatore dinamico perché essendo anch'esso un file (di estensione .ashx) si comporta esattamente come un

file grafico statico per quanto riguarda i riferimenti.
La soluzione consiste nel trasferire il codice del nostro generatore ad una classe all'interno della cartella *App_Code* del nostro sito.
Quindi, in Visual Studio, creiamo un file di codice in *App_Code* chiamandolo *GradientHandler.vb* nel file copieremo il codice di Gradient.ashx inserendo anche un namespace (non è obbligatorio, ma è più elegante):

```
Imports System
Imports System.Web
Imports System.Drawing
Namespace Handlers
    Public Class GradientHandler : Implements
                                            IHttpHandler

        'codice della classe ...

    End Class
End Namespace
```

In questo modo l'*HttpHandler* verrà inserito nella libreria generata in fase di runtime da ASP.NET.
A questo punto, nel file *web.config* dell'applicazione, dovremo inserire il riferimento all'handler in questo modo:

```
<httpHandlers>

        <add path="GradientHandler.ashx" verb="*"
validate="false" type="Handlers.GradientHandler"/>

</httpHandlers>
```

Il fatto di avere impostato l'attributo *validate* su *false* significa che il file non dovrà essere necessariamente presente su disco e che viene richiamata la classe di tipo *Handlers.Gradient Handler* ogni volta che si fa riferimento a un nome di file (virtuale) corrispondente a *GradientHandler.ashx*, qualunque sia la sua posizione.
A questo punto, qualunque sia la posizione del file CSS, sarà sempre possibile usare :
background-image: url('gradient.ashx?ecc… ');
Come ulteriore perfezionamento potremo anche introdurre nel codice il caching delle immagini associando ogni URL alla corrispettiva bitmap da conservare in memoria senza doverla ricostruire ad ogni richiesta ed incrementando così le prestazioni del generatore.

Francesco Smelzo

UB.net

L'AUTORE

Francesco Smelzo è specializzato nello sviluppo in ambiente Windows con particolare riferimento ad applicazioni in ambiente .NET sia web-oriented che desktop. Il suo sito web è www.smelzo.it. **Come sempre è a disposizione per ricevere suggerimenti o richieste sull'articolo all'indirizzo di posta elettronica** francesco@smelzo.it

SVILUPPA PER IL WEB CON YUI

IMPARIAMO A LAVORARE CON IL FRAMEWORK UTILIZZATO DA YAHOO PER SVILUPPARE APPLICATIVI SUL WEB DOTATI DI UNA CERTA COMPLESSITÀ. AL TERMINE DELL'ARTICOLO AVREMO GETTATO LE BASI PER GESTIRE UN CALENDARIO PERPETUO

Lo sviluppo di un sito Web può apparire, al tempo stesso, estremamente semplice o terribilmente complicato, a seconda dei punti di vista: tutto dipende, da numerosi fattori.
Per citarne alcuni:

- innanzitutto, il livello di sicurezza che intendiamo implementare (e questo, naturalmente, apre un ventaglio di problematiche di non poco conto);
- da non trascurare, peraltro, il numero di browser e tecnologie di cui vogliamo assicurare la compatibilità;
- infine, è importante poi considerare il target di riferimento del sito, inteso come l'insieme dei potenziali utilizzatori dello stesso, con le loro esigenze e le loro aspettative.

Se l'impatto grafico non è poi così importante, allora possiamo tranquillamente dedicarci soltanto all'aspetto funzionale; talvolta, però, anche l'occhio vuole la sua parte, soprattutto nei siti che pubblicizzano prodotti commerciali. Un sito "vetrina" deve da un lato essere funzionale, e rispondere con precisione e sollecitudine alle richieste degli utenti, dall'altro essere "piacevole", e spingere i visitatori (anche solo occasionali) ad inserirlo nel bookmark del proprio browser. Ma come fare a rispondere opportunamente a tutte queste esigenze in un colpo solo?
Se il sito è di piccole dimensioni, conviene forse curarne personalmente tanto la forma quanto il contenuto, ma se l'applicazione comincia ad essere particolarmente complessa, allora, conviene sicuramente affidarsi ad un framework specializzato, in grado di velocizzare l'attività di sviluppo senza farci perdere di vista l'obiettivo principale del nostro lavoro, il framework YUI. In questo articolo tratteremo, in particolar modo, le librerie YUI, scaricabili liberamente dal sito della Yahoo (http://developer.yahoo.com/yui/), che raccolgono una serie infinita di funzionalità atte ad agevolare lo sviluppo di siti Web complessi.
Le YUI, peraltro, sfruttano tutta la potenza e l'eleganza del paradigma object-oriented, associato al linguaggio Javascript su cui sono basate.
L'uso di un framework specializzato risolve, infine, un altro annoso problema dello sviluppatore Web: **l'aggiornamento delle proprie applicazioni**. Per tenere un'applicazione sempre aggiornata, infatti, basterà semplicemente (in linea di massima) aggiornare il framework all'ultima versione disponibile, senza toccare, invece, l'"ossatura" del progetto stesso, rappresentata dal codice inserito direttamente dallo sviluppatore.
Il codice di esempio messo a disposizione dalle librerie, oltre a fornire numerose spiegazioni tecniche e un motore di ricerca completo per navigare tutte le funzionalità principali, integra anche strumenti di logging che tengono traccia di tutto ciò che avviene durante le varie fasi di funzionamento dell'applicazione (eventi, errori, tempi di esecuzione del codice, ecc…).

GESTIRE GLI EVENTI CON YUI

Cominciamo ora ad esaminare le classi principali del framework YUI, con un occhio di riguardo, in particolare, alle loro caratteristiche più importanti e utili per lo sviluppatore.
Siccome però il metodo migliore per analizzare un framework complesso come YUI è lavorare direttamente su degli esempi pratici, vi propongo qui di seguito il seguente file:

```
<html>
<head>
<link rel="stylesheet" type="text/css"
href="./yui/build/calendar/assets/calendar.css">
<!-- Dependencies -->
<script type="text/javascript"
src="./yui/build/yahoo-dom-event/yahoo-
                         dom-event.js"></script>
```

```
<!-- Source file and calendar css -->
<script type="text/javascript"
src="./yui/build/calendar/
                              calendar-min.js"></script>
</head>
<body>
<div id="cal1Container"></div>
<script>
var cal1;
function init() {
cal1 = new
    YAHOO.widget.Calendar("cal1","cal1Container");
cal1.render();
}
YAHOO.util.Event.addListener(window, "load", init);
</script>
</body>
</html>
```

Fig. 2: il calendario di YUI in azione

Questo file, presente all'interno dei quickstart di esempio di YUI, inserisce nella pagina web, con poche righe di codice, un calendario perpetuo perfettamente funzionante, che sfrutta unicamente funzioni YUI.
Sorpresi, eh? Niente di magico, tuttavia; semplicemente abbiamo utilizzato una delle funzionalità più interessanti del framework YUI: il controllo **Calendario**, implementato dalla classe **YAHOO.widget.Calendar**. Le classi della utility YAHOO.widget mettono a disposizione dello sviluppatore tutta una serie di controlli davvero interessanti, alcuni dei quali ancora in fase di *beta-testing* (ovviamente, per le mie valutazioni, faccio riferimento alla versione delle librerie più aggiornata fino a questo momento, che è la 2.2.0), altri invece (come appunto il calendario) perfettamente funzionanti e pronti per essere utilizzati nelle nostre applicazioni.
Avremo modo di nuovo, nella prossima puntata, di analizzare meglio altri controlli della utility YAHOO.widget, in questa occasione vogliamo invece soffermarci più che altro sugli **aspetti fondamentali** del loro funzionamento, e più in generale del funzionamento degli oggetti YUI, a cominciare dalla gestione degli **eventi**.
Importando il file yahoo-dom-event.js, posto all'interno del framework, abbiamo a disposizione una serie di funzioni che ci consentono di interagire con facilità con gli eventi DOM della pagina web.
Il file importa inoltre il foglio di stile associato al calendario (dato di default da YUI, ma potremmo linkarne anche uno scritto da noi), e il file che inserisce le funzionalità per la creazione del controllo stesso (calendar-min.js).
L'ultima riga del tag script interno al body, infine, aggancia al caricamento della pagina la funzione **init**, che si limita a creare una nuova istanza dell'oggetto Calendario da posizionare all'interno del div **cal1Container**, stampato poi a video dalla successiva chiamata al metodo **render**.

Per agganciare un gestore di evento al caricamento della pagina, viene utilizzato il metodo **addListener**, il quale definisce una funzione *callback* che viene richiamata automaticamente non appena termina il caricamento della pagina.
La funzione init(), quindi, rappresenta un *event handler* ("gestore di evento") associato all'even-

DEBUGGARE LE APPLICAZIONI WEB

L'immagine qui sotto, visualizza la console di debug:

Questa console si può inserire nella pagina nel seguente modo:

1) innanzitutto, copiamo queste righe nella head della pagina (solo il file logger.js, comunque, è veramente necessario per poter creare l'istanza della console):

```
<script type="text/javascript"
        src="js/yui/build/event/event-
                    debug.js"></script>
<script type="text/javascript"
        src="js/yui/build/logger/logger.js">
                    </script>
```

2) per assegnare uno stile alla console, linkiamo il seguente file CSS:

```
<link type="text/css" rel="style-
  sheet" href="js/yui/build/logger/as
                    sets/logger.css">
```

3) per istanziare il controllo, inseriamo nel body della pagina il seguente codice:

```
var oLogReader = new YAHOO.wid
  get.LogReader(null,{top:"4em",font
    Size:"92%",width:"30em",height:
                          "20em"});
```

E' evidente che questi strumenti velocizzano di gran lunga la realizzazione di siti web complessi, dando al programmatore una piacevole sensazione di "controllo totale" della propria attività di sviluppo, difficilmente ottenibile con altri software disponibili gratuitamente sulla rete.

COS'È FIREBUG?

Alcuni browser sono integrabili con programmi o estensioni che ne arricchiscono notevolmente le funzionalità di base. L'articolo cita, in particolar modo, Firebug, che rappresenta una delle più famose estensioni del browser Firefox, utilissima (assieme ad altre, come ad esempio Web Developer) per supportare l'attività dello sviluppatore Web. Firebug mette a disposizione dello sviluppatore diversi strumenti di debugging del codice Javascript, inoltre fornisce informazioni dettagliate e in tempo reale sulle caratteristiche della pagina corrente, inerenti il codice html, css, le richieste Ajax, ecc...

to load dell'oggetto window. Naturalmente è possibile agganciare più di un event handler ad un oggetto, quando si vuole gestire una molteplicità di eventi per l'oggetto stesso.
I gestori di evento possono poi essere rimossi agevolmente tramite le funzioni **removeListener** (se intendiamo rimuovere uno o più event *handler* specificati) o **purgeElement** (se vogliamo invece rimuovere tutti gli *handler* associati ad un determinato elemento). Quest'ultima funzione, peraltro, dispone di un parametro (di tipo *boolean*, che quindi accetta solo due valori: **true** o **false**) che specifica se la rimozione degli *event handler* deve interessare in modo ricorsivo anche tutti gli elementi di livello inferiore ("figli") a quello preso in considerazione.
Con estrema facilità, quindi, possiamo agganciare alla nostra applicazione web una molteplicità di gestori d'evento, mentre con la stessa disinvoltura è poi possibile rimuoverli o modificarli a seconda delle esigenze particolari del momento.
Osserviamo ora, con particolare attenzione, il semplice script posto all'interno del seguente codice html:

```
<html>
        <head>
                <script>
document.getElementById("testo").value="nuovo_
                                        valore";
                </script>
</head>
<body>
        <input type="text" id="testo"  value="">
</body>
</html>
```

Il browser restituirà, naturalmente, un errore, perché lo script cerca di referenziare un oggetto quando il DOM (Document Object Model) non è ancora disponibile.
Nell'esempio del calendario il problema non si poneva, perché la funzione init() veniva eseguita al termine dell'evento di caricamento, mentre in questo caso lo script, essendo eseguito in anticipo rispetto al codice html nel flusso di caricamento della pagina, si trova a dover lavorare su oggetti che non esistono ancora.
Per evitare questo errore, si potrebbe inserire tale riga di codice subito dopo il tag html che definisce il campo di testo, ma solitamente le funzioni legate al DOM vengono eseguite direttamente su onload della pagina (che scatta solo dopo il caricamento completo del documento, con conseguente perdita di tempo).
Ora, le YUI rendono disponibile la funzione **onAvailable**, facente sempre parte dell'utility **YAHOO.util.Event**, che esegue lo script interessato non appena il DOM dell'oggetto è disponibile. Questo indubbiamente velocizza l'esecuzione del codice all'interno della pagina, mentre lascia la libertà allo sviluppatore di "accentrare" la gestione del DOM anche interamente in un solo file .js, posto magari all'interno della sezione head, senza doversi preoccupare di problematiche legate agli eventi.
Sempre all'interno dell'utility YAHOO.util.Event troviamo, poi, un'altra funzione molto utile: **preventDefault**. Questa funzione, semplicemente, vieta al browser, in occasione del verificarsi di un determinato evento, di comportarsi secondo le sue impostazioni di default (consente, quindi, un maggiore controllo allo sviluppatore che non deve più preoccuparsi dei comportamenti specifici di ogni singolo browser).

MANIPOLARE IL DOM CON YUI

Alzi la mano chi non ha mai avuto problemi con la gestione del DOM in una pagina web di una certa complessità.
Per quanto estremamente utile, infatti, il DOM ha il difetto di generare righe di codice lunghissime, in molte situazioni dove la struttura degli elementi html è abbastanza sofisticata: per aiutarci in queste attività ci viene in aiuto il DOM di YUI.
Innanzitutto, una funzione veramente utile è **getXY**: essa restituisce la posizione X e Y di un oggetto all'interno della pagina web (senza questa funzione dovremmo calcolare manualmente, con un ciclo, la distanza dell'elemento in questione dal vertice superiore sinistro del document, come somma delle varie distanze di ogni singolo elemento rispetto al suo contenitore, fino ad arrivare al body !). Vale la pena, poi, di citare altre due funzioni interessanti: **getClientHeight** e **getClientWidth**, che restitui-

COME POSSO FARE PER INSTALLARE LE YUI ?

**Le librerie YUI sono scaricabili gratuitamente dal seguente sito di Yahoo, http://developer.yahoo.com/yui, dedicato alla comunità degli sviluppatori.
Il progetto YUI, peraltro, facendo parte della comunità opensource, è disponibile anche sul sito http://sourceforge.net/projects/yui.
Per installare il framework è sufficiente scaricarlo da Internet e scompattarlo; a questo punto, quindi, se si vuole integrare il proprio sito con funzionalità YUI basta copiare i file .js interessati e le loro dipendenze direttamente all'interno della cartella del sito stesso.**

scono rispettivamente l'altezza e la larghezza della pagina corrente.
Immaginiamo ora di voler cambiare a run-time la classe CSS associata ad un elemento; in javascript questa compito può essere eseguito dal seguente codice:

```
var
  elemento=document.getElementById("elemento");
elemento.className="nuova_classe";
```

Con queste due righe raggiungiamo con facilità il nostro risultato, tuttavia cosa succederebbe se volessimo ripristinare la situazione che c'era in precedenza ?
Dovremmo memorizzare, per esempio, in un array globale, tutte le varie classi riferite alle diverse situazioni, in modo da poter rintracciare più agevolmente la classe che ci interessa.
Tuttavia sarebbe molto più semplice **aggiungere** la nuova classe a quella già esistente, in modo da limitarci a sovrascrivere gli stili di quest'ultima (senza quindi effettuare realmente una sostituzione): quando poi vogliamo ritornare alla situazione precedente, semplicemente rimuoviamo la classe appena aggiunta.
Le librerie YUI consentono questi "**giochetti**" con una certa facilità; ad esempio, per aggiungere una classe a quelle eventualmente già esistenti, usiamo la seguente istruzione:

```
YAHOO.util.Dom.addClass('contenitore',
                        'modifica_contieni');
```

Firebug (nota estensione del browser Firefox utile per gli sviluppatori web, di cui faccio degli accenni in un box apposito), a questo punto, segnalerà la presenza di più classi CSS sull'elemento in questione, separandole con uno spazio (es. class="modifica modifica_contieni"), mentre visualizzerà gli stili della classe sovrascritta cancellati con una barra orizzontale:

Fig. 3: uno screenshoot di firebug in azione

Per rimuovere la classe appena aggiunta, al contrario, dovremo utilizzare quest'altra riga di codice:

```
YAHOO.util.Dom.removeClass('contenitore',
                           'modifica_contieni');
```

Le funzioni DOM di YUI non si limitano però soltanto a fornire utili strumenti per la manipolazione delle classi CSS: tramite esse possiamo anche svolgere molte altre operazioni, come settare gli stili, visualizzarne le proprietà, ecc...
Un esempio per tutti: per modificare il valore di uno stile, usando la notazione YUI, possiamo operare nel modo seguente:

```
YAHOO.util.Dom.setStyle('content','display','block');
```

Tramite tale riga abbiamo modificato la proprietà **display** dell'elemento di id "**content**", impostandone il valore a "**block**".
Insomma, è evidente che YUI ci aiuta moltissimo nella nostra attività di sviluppo, risparmiandoci spesso i tanti e fastidiosi problemi di compatibilità tra i vari browser, e i "malditesta" che altrimenti dovremmo sopportare per rendere le nostre applicazioni sempre e costantemente aggiornate e al passo con il progresso del Web.

CONCLUSIONI

In questa puntata ci siamo limitati ad introdurre una carrellata di funzioni di utilità, dando solo un piccolo assaggio delle tante e "golose" (in senso metaforico, s'intende !) novità introdotte dalle librerie YUI.
Nel prossimo numero, invece, come anche annunciato più sopra, affronteremo casi pratici, creando finestre, pannelli con eventi associati, interazioni asincrone con server remoti (Ajax), animazioni, ecc…
L'imperativo, quindi, è solo uno: non perdete assolutamente il prossimo numero.

Enrico Viale

L'AUTORE

Enrico Viale è specializzato nello sviluppo di applicazioni sia web-oriented sia desktop. Chi desidera contattarlo per chiarimenti riguardo all'articolo, o per qualsiasi altro motivo, può farlo all'indirizzo enrico.viale@gmail.com.

COME POSSO VELOCIZZARE L'APPLICAZIONE?

Inserire tanti file .js nella sezione head della pagina non migliora, di sicuro, la velocità di download della stessa. YUI mette quindi a disposizione delle versioni "compresse" dei suoi file .js, che terminano con il suffisso *–min*, tuttavia quando i file inclusi sono troppi conviene in ogni caso organizzare diversamente il codice (per esempio, inglobando il codice javascript scritto da noi in un solo file .js invece che distribuirlo su più file), per evitare un decadimento generale delle prestazioni del sito

UN EDITOR HTML NELLE TUE PAGINE

IN QUESTO ARTICOLO IMPAREREMO COME REALIZZARE DELLE FORM MOLTO PARTICOLARI ATTRAVERSO LE QUALI È POSSIBILE INSERIRE IN UN DATABASE TESTI FORMATTATI IN PIENO STILE WORD PROCESSOR. FAREMO TUTTO QUESTO SOLO USANDO JAVASCRIPT

Utilizzando servizi di Web Mail come quelli di Yahoo o Google, vi sarà capitato di incontrare sicuramente il Rich Text Editing, normalmente nella finestra "Nuovo Messaggio", sotto forma di un controllo nella pagina dove l'utente può scrivere del testo formattato (grassetto, corsivo, tipo di carattere, colore ecc…).

Fig. 1: Rich Text Editing di GMail

Ebbene, tali controlli si basano su specifiche funzionalità di alcuni browser (Firefox, Mozilla e Opera) che consentono di impostare il documento, o una area di esso, come editabile dall'utente. Praticamente come le textarea o gli input, ma con la possibilità di formattazione del contenuto ottenuta con l'HTML. Bisogna precisare che queste funzionalità non sono codificate come standard e quindi ogni browser fornisce un approccio talora anche molto diverso al problema.

ENTRARE IN MODALITÀ EDITING CON IE E OPERA

Internet Explorer ha l'indubbio merito di aver introdotto, fin dalla versione 5, le funzionalità di editing integrate nel browser e presenta a tutt'oggi, a mio avviso, la migliore implementazione.
Il browser di Microsoft consente di impostare due modalità di editing :

1. La possibilità di rendere editabile l'intero documento impostando la proprietà designMode dell'oggetto document su "On".
2. La possibilità di rendere editabili solo alcuni elementi della pagina (come ad esempio i tag DIV) o impostandone l'attributo CONTENTEDITABLE su "true" oppure impostando, attraverso il DOM, la proprietà contentEditable, sempre su true, dell'elemento.

Nel primo caso occorrerà impostare la proprietà designMode nell'evento onLoad:

```
<html>
<head></head>
<body onload="document.designMode='On'">
</body>
</html>
```

Nel secondo caso possiamo impostare l'attributo sull'elemento:

```
<html>
<head></head>
<body>
        <div id="editableArea"
                contentEditable="true"></div>
</body>
</html>
```

oppure farlo a livello di script:

```
<html>
<head>
<script language="javascript"
                type="text/javascript">
        window.onload = function () {
document.getElementById("editableArea").contentE
                ditable=true;
        }
</script>
</head>
<body>
        <div id="editableArea"></div>
</body>
```

```
</html>
```

ENTRARE IN MODALITÀ EDITING CON FIREFOX

Firefox, a differenza di IE e Opera, non consente l'editabilità di un singolo elemento della pagina, ma ha soltanto la possibilità di rendere editabile l'intero documento, sempre attraverso l'impostazione su on della proprietà *designMode* dell'oggetto *document*, come abbiamo visto nel primo esempio.
Questo in pratica ci costringe a dedicare all'editor una *Frame* o *IFrame* separata nella pagina per avere la possibilità di avere una *Toolbar* esterna all'area editabile (altrimenti sarebbe editabile anch'essa).

METTERE D'ACCORDO TUTTI

Volendo sviluppare una soluzione *cross-browser* non ci resta quindi che tenere conto delle limitazioni di Firefox e inserire la parte editabile in *Frame* o *IFrame* separate. Procediamo dunque a sviluppare un piccolo esempio per fissare meglio i concetti. In una directory, quindi creeremo due file HTML:

1. il primo, chiamiamolo *esempio.html*, conterrà un'*IFrame*.
2. il secondo, *editor.html*, occuperà l' *IFrame*.

Quindi il codice di esempio.html, sarà pressappoco :

```
<html>
<meta http-equiv="content-type"
        content="text/html; charset=ISO-8859-1"/>
<head>
</head>
<body>
        <div id="Toolbar"></div>
        <iframe id="edit" width="100%"
    height="200" marginheight="0" marginwidth="0"
                src="editor.html"/>
```

FCKEDITOR

Per chi ha bisogno di un editor HTML affidabile già pronto è senz'altro consigliabile l'ottimo FCKEditor (www.fckeditor.net), un progetto Open Source di Frederico Caldeira Knabben. FCKEditor si può integrare con i più diffusi linguaggi lato server (ASP,ASP.NET, Perl, PHP, Python ed altri ancora) oppure può essere usato lato client. Implementare FCKEditor richiede un po' di apprendimento anche perché la sua struttura è molto ricca di funzionalità ed anche estendibile attraverso plugin. Vediamo, in estrema sintesi, un'implementazione di base.

INSTALLAZIONE

Scarichiamo i file della libreria e decomprimiamo la directory contenente tali file nel sito da sviluppare (ad esempio C:\miosito\editor).
Nel codice HTML della pagina che deve ospitare l'editor inseriamo il riferimento alla libreria rappresentata dal file fckeditor.js :

```
<head>
        <title>FCKeditor - Sample</title>
        <script type="text/javascript"
        src="editor/fckeditor.js"></script>
</head>
```

UTILIZZO

A questo punto è sufficiente instanziare l'editor nel corpo del documento HTML con un semplice script

```
<form action="sampleposteddata.asp"
        method="post" target="_blank">
                <script type="text/java
                        script">
<!--
var oFCKeditor = new FCKeditor( 'FCKeditor1' )
                                        ;
oFCKeditor.BasePath =
                document.location.pathname;
oFCKeditor.Height    = 300 ;
oFCKeditor.Value     = 'This is some
        <strong>sample text<\/strong>. You are
                                using <a
        href="http://www.fckeditor.net/">FCKedi
                                tor<\/a>.' ;
oFCKeditor.Create() ;
//-->
                </script>
                <br />
                <input type="submit" va
                        lue="Submit" />
</form>
```

Come notiamo sono stati impostati alcuni parametri dell'oggetto *FCKeditor*:

- **BasePath – la directory corrente, relativa a dov'è collocata la libreria**
- **Height – altezza del controllo in pixel**
- **Value – il testo vero e proprio**

Infine viene richiamato il metodo *Create()* per costruire il codice HTML necessario. Se tutto è andato bene la pagina si dovrebbe presentare come vediamo in figura A, con l'editor in bella mostra

DOCUMENTAZIONE

Naturalmente nella vita reale le cose sono ben più complicate, a cominciare con l'integrazione con il lato server e la configurazione della libreria; ci viene però in aiuto il sito di documentazione del progetto (http://wiki.fckeditor.net) abbastanza dettagliato.

LIMITI

Naturalmente ogni medaglia ha il suo "rovescio", quello di FCKEditor è che proprio a causa della ricchezza di funzionalità, la libreria risulta un po' pesante da caricare (addirittura anche su connessioni locali); questo la rende adatta ad essere utilizzata più su ambienti Intranet che sul Web vero e proprio, e comunque è certamente da evitare l'uso nelle pagine chiave del sito come la Home.

```
</body>
</html>
```

mentre *editor.html*, sarà:

```
<html>
<head></head>
<body contentEditable="true"
                    onload="document.designMode='On'">
</body>
</html>
```

in modo da funzionare sicuramente con i tre browser. Se adesso apriamo con uno dei browser (per Opera occorre la versione 9) la pagina *esempio.html* noteremo che nella IFrame (quella dell'*editor*) possiamo scrivere liberamente. Fin qui nulla di molto diverso da una comune *textarea*, dovremo infatti occuparci del modo in cui impostare la formattazione del testo.

LA FORMATTAZIONE DEL TESTO

Per fortuna la formattazione del testo è, concettualmente simile, tra i vari browser: eseguire il metodo execCommand su document o su un area selezionata. execCommand accetta i seguenti parametri:

- Nome del comando – indica appunto la stringa identificativa del comando (come poi vedremo)
- Flag interfaccia utente – indica se attivare l'interfaccia Utente, ad esempio una finestra di dialogo per i colori ecc…, in pratica però questo parametro viene sempre settato a false perché il comportamento dell'interfaccia non è uniforme in tutti i browser.
- Valore – il valore da passare, se necessario, al comando; se non viene utilizzato va impostato a null.

Vediamo quindi, una tabella sinottica, quali sono i principali comandi che possono essere impartiti. L'applicazione dei comandi avviene appunto richiamando il metodo *execCommand* sull'oggetto document della *Frame* o *IFrame* che ospita il documento editabile:

```
<script language="javascript"
```

COMANDO	VALORE	COMPORTAMENTO	IE	FF	OP
backColor	colore	In IE imposta lo sfondo del testo selezionato, in Firefox e in Opera imposta lo sfondo del documento	X	X	X
bold	Nessuno	Imposta o rimuove la formattazione in grassetto del testo selezionato	X	X	X
copy	nessuno	Copia il testo selezionato negli appunti di sistema	X*	X*	X*
createBookmark	nome anchor	Inserisce un Bookmark intorno al testo selezionato con il nome passato come valore	X		
createlink	url (href)	Inserisce un link intorno al testo selezionato all'url passato come valore	X	X	X
cut	nessuno	Taglia il testo selezionato e lo mette negli appunti di sistema	X*	X*	X*
decreasefontsize	nessuno	Aggiunge il tag <small> intorno al testo selezionato riducendone le dimensioni		X	X
delete	nessuno	Cancella il testo e gli oggetti selezionati	X	X	X
fontname	nome	Imposta il nome del carattere della selezione	X	X	X
fontsize	valore da 1 a 7	Imposta la grandezza del carattere	X	X	X
forecolor	Colore	Imposta il colore del testo	X	X	X
Formatblock	<h1><h2> ecc…	Imposta lo stile del blocco di testo selezionato	X	X	X
Heading	<h1><h2> ecc…	Simile a formatblock, ma solo per gli stili Heading	X	X	X
increasefontsize	nessuno	Aggiunge il tag <big> intorno al testo selezionato aumentandone le dimensioni		X	X
indent	nessuno	Indenta il blocco di testo dov'è posizionato il cursore.	X	X	X
inserthorizontalrule	nessuno	inserisce una linea orizzontale (<hr>)	X	X	X
inserthtml	stringa html	Inserisce del codice HTML dov'è posizionato il cursore.		X	X

COMANDO	VALORE	COMPORTAMENTO	IE	FF	OP
insertimage	url (src)	Inserisce un'immagine (dall'url passato) dov'è posizionato il cursore.	X	X	X
insertorderedlist	nessuno	Inserisce o elimina una lista numerata	X	X	X
insertunorderedlist	nessuno	Inserisce o elimina una lista puntata	X	X	X
insertparagraph	nessuno	Inserisce un paragrafo <P>	X	X	X
italic	nessuno	Imposta o rimuove la formattazione in corsivo del testo selezionato	X	X	X
justifycenter, justifyfull, justifyleft, justifyright	nessuno	Imposta l'allineamento del paragrafo corrente	X	X	X
outdent	nessuno	Fa rientrare una precedente indentazione del testo	X	X	X
paste	nessuno	Incolla testo dagli appunti	X*	X*	X*
redo	nessuno	Ripristina ciò che è stato in precedenza annullato con il comando *UNDO*		X	X
removeformat	nessuno	Rimuove la formattazione	X	X	X
Selectall	nessuno	Seleziona tutto	X	X	X
strikethrough	nessuno	Imposta la formattazione di testo attraversato da linea		X	X
styleWithCSS	true false	Usa i CSS per i comandi di formattazione anziché i tag HTML		X	X
underline	nessuno	Aggiunge il tag <u> intorno al testo selezionato facendolo apparire sottolineato	X	X	X
undo	nessuno	Annulla un azione precedente		X	X
unlink	nessuno	Rimuove un link	X	X	X

** = in alcuni browser ci possono essere restrizioni di sicurezza per i comandi specificati*

```
type="text/javascript">
function applyBold() {
var iframe = document.getElementById("edit");
iframe
    .contentWindow.document.execCommand("bold",fals
e,null);
}
</script>
...
<iframe id="edit">
```

UNA PRIMA APPLICAZIONE

Vediamo adesso come mettere insieme quello che abbiamo visto in una prima, semplice, applicazione di Rich Text Editing. I file che abbiamo visto in precedenza, lo ricordiamo, rappresentano il contenitore (esempio.html) e il documento editabile contenuto nell'IFrame (editor.html). Il nostro scopo sarà quello di avere una Toolbar con i fondamentali comandi di formattazione:

- Grassetto
- Corsivo
- Sottolineato
- Allineamento a sinistra
- Allineamento centrato
- Allineamento a destra
- Elenco numerato
- Elenco puntato

Definiamo quindi l'HTML del contenitore (esempio.html) così:

```
<html>
<head>
        <title>Editor Demo</title>
        <script language="javascript"
                        type="text/javascript">
</script>
</head>
<body>
        <div id="Toolbar"></div>
        <iframe id="edit" src="editor.html"
           width="100%" height="200px"></iframe>
        </body>
</html>
```

dove:

- L'elemento DIV con id Toolbar è quello che ospi-

terà la nostra tabella degli strumenti.

- L'elemento IFRAME con id edit è la finestra che contiene il testo editabile.

Quindi, per prima cosa, definiamo, in javascript, un array di comandi espressi come oggetti:

```
var commands = [
        {name:'bold',arg:null,icon:'bold'},
        {name:'italic',arg:null,icon:'italic'},
        {name:'underline',arg:null,icon:'underline'},
        {name:'sep'},
        {name:'justifyleft',arg:null,icon:'alignl'},
        {name:'justifycenter',arg:null,icon:'alignc'},
        {name:'justifyright',arg:null,icon:'alignr'},
        {name:'sep'},
{name:'insertorderedlist',arg:null,icon:'numlist'},
{name:'insertunorderedlist',arg:null,icon:'bullets'}
];
```

dove *name* sarà il comando da richiamare, *arg* il valore da passare (per questi comandi sempre *null*) e icon il nome del file della GIF che rappresenta il comando.
Passiamo quindi a definire la funzione di costruzione della *Toolbar*, attraverso una tabella generata dinamicamente che contiene i pulsanti:

```
function setupToolbar (){
        var ToolbarTable =
                    document.createElement("table");
        var ToolbarRow = ToolbarTable.insertRow(-
                                                    1);
        for(var i=0;i<commands.length;i++){
                var ToolbarCell =
                        ToolbarRow.insertCell(-1);
                var command = commands[i];
                if(command.name!="sep") {
                        ToolbarCell.innerHTML=
      '<img src="images/' + command.icon  + '.gif" >';
                        ToolbarCell.onclick =
     new Function ('exec("' + command.name + '")');
                }
                else ToolbarCell.innerHTML=
                                            ' ';
        }
document.getElementById("Toolbar").appendChild
                                    (ToolbarTable);
}
```

come potete osservare utilizziamo l'array di comandi per definire il contenuto delle celle e l'evento onclick sul pulsante che punta alla funzione *exec* ed infine appendiamo il nodo *ToolbarTable* all'elemento *Toolbar* che abbiamo definito nel codice HTML.
La funzione *exec* prende semplicemente il nome del comando e lo passa ad *execCommand* dell'oggetto document della finestra dell'editor:

```
function exec (name){
        var editor =
                    document.getElementById("edit")
editor.contentWindow.document.execCommand(name
                                    ,false,null);
}
```

Il terzo parametro della funzione *execCommand* è sempre impostato a *null* perché i comandi che abbiamo preso in considerazione non richiedono un valore.
Infine, nell'evento *onload* di *window* ci assicuriamo che la proprietà *designMode* del documento editabile sia su "on" e richiamiamo la funzione *setupToolbar*:

```
window.onload = function () {
        var doc =
            document.getElementById('edit').contentWindow.
document;
        doc.designMode = "on";
        setupToolbar();
}
```

Alla fine del nostro lavoro apriamo la pagina in nei tre browser è potremmo vedere il nostro mini-editor in azione (figura 2).

Fig. 2: Rich Text Editor su strada

Certo, non è ancora paragonabile a Word, tuttavia è già un primo passo…

ALTRI ELEMENTI TECNICI

Naturalmente per creare un vero e proprio editor completo di tutte le funzionalità manca ancora qualcosa. Nel nostro esempio, per non disperderci inutilmente in troppi dettagli implementativi, abbiamo visto solo i comandi più semplici, c'è però da prevedere la possibilità di:

- impostare il colore del testo
- impostare il font del testo

DOCUMENTAZIONE

La documentazione e i riferimenti sul Rich Text Editing possono essere reperite in rete: per IE in http://msdn.microsoft.com/workshop/author/editing/editing_entry.asp per Firefox in http://www.mozilla.org/editor/midas-spec.html per Opera la documentazione non è certo abbondantissima, comunque il sito di riferimento è http://dev.opera.com

- impostare le dimensioni del testo
- impostare e rimuovere dei link
- impostare immagini

Comunque i concetti di base non cambiano, soltanto che qui dovremo prevedere meccanismi più sofisticati di interazione con l'utente, come finestre di dialogo, pop-up ecc…
Un altro *task* importante da implementare sarebbe la sensibilità al contesto della toolbar, mi spiego meglio, sarebbe opportuno che quando il cursore si trova sopra un elemento formattato, ad esempio in grassetto, il relativo bottone venisse in qualche modo evidenziato, un po' come succede in Word.
Il comando per controllare lo stato della formattazione del testo su cui insiste il cursore è *queryCommandValue* dell'oggetto *document* della finestra editabile.
Il comando agisce come *execCommand*, anch'esso accetta come parametro il nome del comando, solo che restituisce lo stato della formattazione.
Ad esempio :

```
document.queryCommandValue("bold");
```

restituirà true/false a seconda della formattazione corrente, o ancora

```
document.queryCommandValue("fontName");
```

restituirà il nome del carattere della formattazione corrente. E così via.
Quindi, usando *queryCommandValue*, si potrebbe innescare, su *onmouseup* e *onkeyup* sulla finestra dell'editor, un meccanismo di messaggi alla Toolbar per settare lo stato dei vari controlli:

```
window.onload = function () {
        var win =
        document.getElementById('edit').contentWindow
        var doc = win.document;
        …
        doc.onmouseup = doc.onkeyup =
                                editorChange;
}
//comandi da monitorare
var queryCommands =
    ['bold','italic','underline','justifyleft','justifycenter','just
                                ifyright'];

//funzione di aggiornamento
function editorChange (){
        var editorDoc =
    document.getElementById("edit").contentWindow.doc
                                ument;
        for(var i=0;i<queryCommands.length;i++){
                var
                queryCommand=queryCommands[i];
                var b =
        editorDoc.queryCommandValue(queryCommand);
                var cmdButton =
    document.getElementById("btn" + queryCommand);
                if(!b)
                    cmdButton.className="toolbarButton";
                else
                    cmdButton.className="toolbarButtonOn";
}
}
```

INTEGRAZIONE CON IL LATO SERVER

Lo scopo principale del nostro Rich Text Editor sarà ovviamente quello di comunicare con il server per processare le informazioni immesse dall'utente.
Per far questo occorre naturalmente, per prima cosa, recuperare il codice HTML generato dalla formattazione del testo. Per farlo è sufficiente far riferimento alla proprietà innerHTML del dell'oggetto *document.body* della finestra editabile, in questo modo :

```
function save (){
        var editorDoc =
    document.getElementById("edit").contentWindow.doc
                                ument;
        var code =editorDoc.body.innerHTML;
        alert(code);
}
```

naturalmente in questo esempio il codice viene solo mostrato in una finestra di dialogo, mentre in un caso reale dovrà essere inviato al server con un POST nascosto o con XMLHttpRequest.

PERCHÉ SVILUPPARE UN RICH TEXT EDITOR?

Nel web ci sono già da tempo diversi editor di questo tipo, perché quindi la fatica di svilupparlo da soli Per due ottimi motivi:

1. conoscere la tecnica che sta alla base del Rich Text Editing nei vari browser ci consente di avere un maggior controllo anche nell'uso di librerie sviluppate da altri e anche, perché no, aggiungere funzionalità mancanti.

2. le librerie disponibili sul web, dovendo giustamente prevedere numerosi casi d'uso sono per forza di cosa "generaliste", E possono raggiungere una complessità e una pesantezza che non ci occorre

Enrico Viale

NOTA

RISORSE IN RETE

Per chi proprio non volesse "fare da solo" oltre all'ottimo FCKEditor (solo per IE e Firefox) di cui abbiamo già parlato c'è TinyMCE che copre tutti i Browser (IE, Firefox, Opera e Safari) scaricabile all'indirizzo http://tinymce.moxiecode.com.
Una lista particolarmente ricca dei Rich Text Editor disponibili è invece su http://www.geniisoft.com/showcase.nsf/WebEditors

ALLA SCOPERTA DEGLI INTERCEPTOR

ALLA NOSTRA APPLICAZIONE È GIUNTA UNA RICHIESTA, POSSIAMO GESTIRLA FACILMENTE CON UNA ACTION, MA COME FARE SE VOLESSIMO "MANIPOLARLA" PRIMA CHE ESSA SIA GESTITA? LA RISPOSTA È CONTENUTA IN QUESTO ARTICOLO

Le Action, che sono già state ampiamente analizzate nei precedenti articoli dedicati al nuovo framework, racchiudono sostanzialmente la componente controller del design pattern MVC implementato da Struts 2. È difatti nella action che si invocano i metodi di business logic, si raccolgono eventuali risultati e si seleziona la vista a cui demandare la responsabilità di visualizzarli.
Gli interceptor, invece, sono degli oggetti che "circondano" ogni action, elaborando la richiesta prima che questa giunga alla stessa ed elaborando la risposta generata. Un interceptor, oltre a racchiudere una action, può essere posizionato in modo da elaborare la richiesta e la risposta rispettivamente passate e restituite da un altro interceptor a valle. Questa possibilità che è la vera caratteristica saliente degli interceptor offre la possibilità di costruire catene che possono modificare anche in modo notevole il comportamento di Strut 2.
Durante i primi esperimenti con Struts 2 non ci si accorge neanche della presenza degli interceptor ma questi, a ragione, ne rappresentano la vera anima in quanto tutte le comodissime funzionalità fornite dal framework tra le quali citiamo la localizzazione con la sostituzione automatica dei messaggi tradotti e la validazione dei campi delle form con l'eventuale ripresentazione dei dati errati e dei messaggi esplicativi, sono funzionalità implementate attraverso gli interceptor.
Struts 2 prevede una serie di default di interceptor che vengono applicati ad ogni action che non specifichi nessuna configurazione particolare. Il cosiddetto stack di default prevede ben 17 interceptor che lavorano dietro le quinte per offrire vari servizi. Tra i tanti interceptor ricordiamo i seguenti:

- Exception: mappa una particolare eccezione al nome di una vista.
- Prepare: richiama un metodo di inizializzazione della Action nel caso questa implementi una determinata interfaccia.
- I18n: gestisce la memorizzazione del locale per l'utente corrente.
- Debugging: permette di attivare il debug delle viste.
- FileUpload: per una gestione più semplice dell'upload dei file
- Validation: esegue la validazione dei dati forniti nella form congruentemente al contenuto dei relativi file xml di definizione dei controlli.

Oltre agli interceptor inclusi nello stack di default ne esistono molti altri non inclusi perché utili in situazioni particolari e non in ogni caso. In questo articolo vedremo come utilizzare uno degli interceptor messi a disposizione da Struts 2: waitAndExecute.

IL PROBLEMA

Durante questa serie di articoli abbiamo sviluppato un'applicazione dedicata al mondo immobiliare attraverso la quale sostanzialmente un utente è in grado di cercare tra gli immobili offerti dalla agenzia quelli venduti ad un prezzo compreso in una determinata fascia.
Negli esempi finora mostrati la ricerca veniva portata a termine in un tempo irrisorio. Nel caso però ci si dovesse riferire ad una ricerca più completa su database, o attraverso qualche web service, la ricerca avverrebbe sicuramente in un tempo maggiore.
In un tale scenario sarebbe controproducente far attendere l'utente sino a che la pagina con i risultati non venga visualizzata. Molto meglio sarebbe visualizzare una pagina di cortesia che informi il possibile acquirente che la ricerca è in corso, e magari sia anche in grado di fornire graficamente una stima del tempo di attesa rimasto, magari attraverso una progress bar.
L'interceptor "executeAndWait" di Strut 2 è proprio stato pensato con questo scenario in mente. Sostanzialmente questo interceptor ese-

gue una action in background e attende per un certo tempo che la stessa termini. In questo caso il risultato viene passato alla vista selezionata dalla action e quindi l'interceptor risulta totalmente ininfluente. Se però la action non termina nel tempo stabilito l'interceptor passa alla vista "wait" rendendo disponibile alla stessa proprio la action che intanto è in esecuzione. Non rimane altro da fare che realizzare la vista "wait" in modo che tramite un reload ricaricarichi la pagina dei risultati. L'interceptor continuerà a reindirizzare la richiesta sulla pagina di attesa sino a quando la action non sarà temrinata e i dati diverranno disponibili.

Fig. 1: La finestra di ricerca non ha subito modifiche per l'inserimento dellàinterceptor.

SIMULAZIONE DI UNA ACTION CON LUNGHI TEMPI DI ESECUZIONE

La Action di ricerca Search offre il metodo getFoundProposals() che sarà invocato all'interno della vista per ottenere i risultati richiesti.
Nel metodo execute() viene richiamato il metodo slowFind() che simula la chiamata ad un DAO che ha la responsabilità di fornire i risultati. I metodi setProposalsToCheck() e setCheckedProposals() permettono ad un oggetto di impostare il numero di proposte che saranno restituite e quanti oggetti sono stati effettivamente caricati. Tramite questi due parametri la action è in grado di fornire un indice di completamento del lavoro facendo un semplice rapporto tra le due quantità. Questo risultato e riportato dal metodo getProgress() e potrà variare tra 0 (esecuzione appena iniziata) e 1 (esecuzione completata).

```
package com.danidemi.realestate;
...
public class Search  extends ActionSupport {
        private Integer minValue;
        private Integer maxValue;
        private Set<Proposal> foundProposals;
```

```
        private int proposalsToCheck = 0;
        private int checkedProposals = 0;
        public Integer getMaxValue() {...}

        public void setMaxValue(Integer maxValue)
                                                {...}

        public Integer getMinValue() {...}

        public void setMinValue(Integer minValue)
                                                {...}
        public String execute() throws Exception {
                if(minValue > maxValue){
                        addActionError("Il
valore minimo deve essere inferiore a quello
                                        massimo.");
                        return INPUT;
                }

                Portfolio portfolio =
                                Portfolio.instance();
                foundProposals =
                                portfolio.slowFind(this);
                return SUCCESS;
        }
        public Set<Proposal> getFoundProposals()
                                                {...}
        public void setProposalsToCheck(int i) {
                this.proposalsToCheck = i;
        }
        public void setCheckedProposals(int
                                        checked) {
                this.checkedProposals = checked;
        }
        public double getProgress(){
                double progress = 0;
                if(proposalsToCheck!=0){
                        progress =
                checkedProposals / proposalsToCheck;
                }
                return progress;
        }
        public void resetProgress() {
                setCheckedProposals(0);
                setProposalsToCheck(0);
        }
}
```

ESEGUIRE LA RICERCA

La ricerca degli immobili che soddisfano i parametri impostati deve avvenire più lentamente. I ritardi dovuti ad esempio ad una accesso ad un database sono qui simulati inserendo una chiamata a Thread.sleep() per ogni proposta che viene caricata. In questo modo si cerca di palesare il comportamento dell'interceptor. Per ogni proposta caricata

viene inoltre richiamato il metodo setCheckedProposal() in modo che la ricerca possa calcolare di volta in volta la percentuale di lavoro completato.

```
public Set<Proposal> slowFind(Search search){
    HashSet<Proposal> result = new HashSet<Proposal>();
    search.resetProgress();
search.setProposalsToCheck(portfolio.size());
    int checked = 0;
    for (Proposal proposal : portfolio) {
        Thread.yield();
        search.setCheckedProposals(checked);
if(proposal.isValueInRange(search.getMinValue(), search.getMaxValue())){
result.add(proposal);
        }
        checked++;
        try {
Thread.sleep(400);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
e.printStackTrace();
        }
    }
    return result.isEmpty() ? null : result;
}
```

Fig. 2: Ecco come viene visualizzata la pagina di attesa nello stato iniziale.

CONFIGURAZIONE DELL'INTERCEPTOR

L'interceptor viene configurato all'interno della definizione della action nel file struts.xml.
Per fare ciò si usa il tag interceptor-ref che si riferisce ad interceptor o catene di interceptor preconfigurate.
In questo caso il primo tag si riferisce allo stack di default di Struts 2. Questo viene in realtà "montato" attorno all'interceptor execAndWait che si occupa di selezionare la vista di attesa nel caso la action non termini per tempo. Come abbiamo già visto nei precedenti articoli una vista è sostanzialmente definita da un nome. L'interceptor execAndWait restituisce "wait" nel caso la Action che sottintende non termini per tempo, altrimenti restituisce l'identificativo restituito dalla action stessa. Va da se quindi che dobbiamo prevedere una nuova pagina nel caso la vista selezionata sia "wait".
Il parametro "delay" indica i millisecondi per i quali l'interceptor attenderà prima di selezionare la vista "wait".

```
<action name="Search"
    class="com.danidemi.realestate.Search">
    <interceptor-ref name="completeStack"/>
    <interceptor-ref name="execAndWait">
        <param
            name="delay">1000</param>
    </interceptor-ref>
    <result
        name="success">/Hello.jsp</result>
    <result
        name="input">/Index.jsp</result>
    <result name="wait">/Wait.jsp</result>
</action>
```

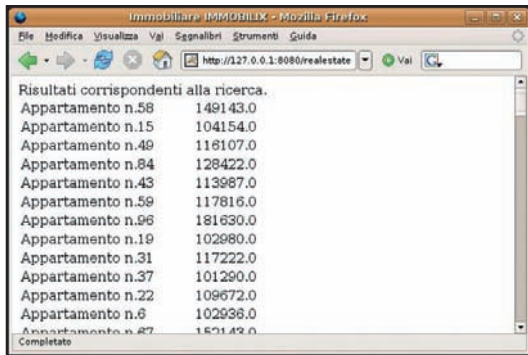
LA PAGINA DI ATTESA

La pagina di attesa e realizzata in modo che dopo 5 secondi provi a rieseguire la action di cui l'utente sta attendendo il completamento. Per fare ciò viene utilizzato il meta tag "refresh" che automaticamente carica un URL dopo un tempo predefinito di secondi. L'URL da caricare è fornito tramite il tag "url" delle tag library struts ed è quello utilizzato per l'invocazione della action stessa. Si noti che nella pagina sono state introdotti due nuovi messaggi localizzati: "wait.message" e "wait.reload Message", il primo per avvertire l'utente che la action invocata è ancora in esecuzione, e il secondo per dare la possibilità di ricaricare la pagina nel caso il browser non dovesse sopportare il meta tag refresh. Si noti come per accedere alla percentuale di progresso della action venga usato il tag "property" delle tag library Struts 2. Questo è possibile perché l'interceptor executeAndWait, come specificato nella relativa documentazione, pone la action in esecuzione in cima allo StackValue. Lo StackValue e una struttura dati a pila che viene passata ad ogni pagina e che contiene i bean da essa accedibili. Quando ci si riferisce semplicemente ad una property attraverso il suo nome, Struts tenta di risolverla tra le proprietà del primo oggetto dello stack, che in questo caso è

proprio l'action in esecuzione. Il metodo invocato sarà quindi getProgress(). Per dare un'idea della progress bar viene semplicemente estratta la percentuale di completamento e attraverso un'opportuna moltiplicazione si genera l'immagine della progress bar come allungamento di un singolo pixel verde (per la percentuale eseguita) e di un singolo pixel grigio (per la percentuale di lavoro ancora da eseguitre).

Fig. 3: Far attendere l'utente del sito la generazione della pagina con tutti i risultati sarebbe stato controproducente.

```
<s:i18n name="RealEstate">
        <html>
        <head>
                <meta http-equiv="Content-
        Type" content="text/html; charset=UTF-8">
                <meta http-equiv="refresh"
    content='5;url=<s:url includeParams="all"/>'/>
                <title>
                        <s:text
                    name="site.title"></s:text>
                </title>
        </head>
        <body>
                <s:text name="wait.message"/>
                <a href='<s:url
                  includeParams="all"/>'><s:text
name="wait.reloadMessage"/></a>
                <s:property value="progress"
                                          />
                <table align="center"
                                border="1">
                        <tr>

                                <td>

<img src="images/pxgreen.gif" height="20"
    width='<s:property value="300 * progress"/>'>
<img src="images/pxgrey.gif" height="20"
                width='<s:property value="300 *
                           (1 - progress)"/>'>
                                        </td>
                        </tr>
```

```
                        </table>
                </body>
                </html>
</s:i18n>
```

COMPLETARE I FILE DI LOCALIZZIAMONE

Dato che ci sono nuovi messaggi localizzati è necessari apportare le opportune modifiche ai file di localizzazione. Qui di seguito riportiamo la modifica al solo file RealEstate_it.properties che contiene le traduzioni italiane. Le stesse chiavi andranno aggiunte agli altri due file, quello generico e quello inglese.

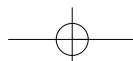
```
wait.message=Ricerca in corso. Attendere.
wait.reloadMessage=Clicca qui se la pagina non si
                        ricarica entro 5 secondi.
```

TEST FINALE CON UN BROWSER

Avviate ora l'applicazione web portandovi nella directory bin della vostra installazione di Tomcat e lanciando il comando “catalina start”. Nel caso non abbiate variato nessun parametro rispetto ai sorgenti forniti l'applicazione dovrebbe rispondere all'URL http://127.0.0.1:8080/realestate/Index.jsp. Inserite due valori, minimo e massimo, compresi tra 100000 euro e 200000 euro, visto che tutte le offerte generate sono comunque in questa fascia di prezzo. Premete il pulsante per avviare la ricerca e ora dovreste visualizzare la progress bar.

CONCLUSIONI

Con questo articolo termina questa breve rassegna su Struts2. Abbiamo cercato di mostrare attraverso ripetuti sviluppi di un'applicazione esemplificativa come applicare alcune tra le più interessanti caratteristiche del framework, che ovviamente offre numerosissime potenzialità come ad esempio la programmazione ed il deploy di nuovi interceptor. Struts2 si avvia sicuramente ad essere una piattaforma più completa e flessibile della precedente versione. Il consiglio è sicuramente quello di partire con la nuova versione nel caso si debba realizzare un nuovo progetto partendo da zero e non ci siano quindi problemi di passaggio da una tecnologia all'altra

Daniele De Michelis

LA SICUREZZA DI WINDOWS XP

GLI HACKER SONO IN AGGUATO PRONTI A SFRUTTARE QUALSIASI VULNERABILITÀ DI WINDOWS XP E DELLE SUE APPLICAZIONI : ECCO PERCHÉ È DI FONDAMENTALE IMPORTANZA CONOSCERE IL SISTEMA DI PROTEZIONE CHE MICROSOFT RENDE DISPONIBILE.

I sistemi operativi Microsoft vengono frequentemente additati a causa della loro scarsa sicurezza e della presunta facilitá con cui gli hacker potrebbero forzarne le difese. Se da un lato è innegabile che spesso vengono rilasciati software con bug di security che solo successivamente vengono eliminati tramite patch, è altrettanto vero che chi ne ha la responsabilitá, a livello domestico o aziendale, dovrebbe seguire linee guida precise a partire dalla fase d'installazione e utilizzano cautele tali da evitare che malintenzionati siano in grado di attaccarli e manipolarli facilmente. Prima di parlare di "software good enough", come spesso i detrattori di Microsoft sono abituati a fare, bisognerebbe essere certi che gli amministratori di sistema siano sufficientemente preparati sulla regole della sicurezza ed in grado di portare avanti azioni proattive e non solo reattive a danno avvenuto.
In effetti, anche se sono state installate tutte le patch di sicurezza raccomandate, non è possibile pensare di essere al sicuro da possibili attacchi. Gli hacker sono in agguato pronti a sfruttare qualsiasi vulnerabilità del sistema operativo e delle applicazioni che girano al suo interno: ecco perché è di fondamentale importanza essere informati che il sistema di protezione di Windows XP rende disponibili importanti strumenti per analizzare lo stato attuale della sicurezza, interpretare i risultati ottenuti e, di conseguenza, effettuare le correzioni dovute.

L'SCM può essere utilizzato da un'applicazione GUI che viene eseguita come snap-in all'interno della Microsoft Management Console. Tuttavia quando è necessario applicare file di configurazione a svariati sistemi è conveniente fare ricorso a secedit.exe, eseguito direttamente sulla linea di comando.

Area	Configurable Items
Account Policies	• Password Policy • Account Lockout Policy
Local Policies	• Audit Policy • User Rights Assignments • Security Options (Registry Values)
Event Log	• Settings for System, Application, and Security Logs
Restricted Groups	• Group membership
System Services	• Startup Modes and Access Control Lists for all system services
Registry	• Access Control Lists for Registry Keys
File System	• Access Control Lists for Folders and Files

***Figura 1:** Aree dove è possibile utilizzare il Security Configuration Manager per configurare le impostazioni relative alla sicurezza*

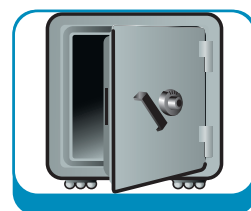
Tenete presente, tuttavia, che il Security Configuration Manager non è uno strumento che agisce direttamente sulla configurazione attuale del sistema in oggetto, ma serve ad attivare un esercizio di verifica e di analisi delle impostazioni correnti rispetto ad uno standard ottimale. Al termine di questo esercizio, grazie alle indicazioni ora disponibili, si potranno applicare i nuovi standard di sicurezza, che da quel momento in poi saranno validi per il sistema o i sistemi di riferimento.

IL SECURITY CONFIGURATION MANAGER

Lo strumento fondamentale per la gestione e l'analisi della configurazione della sicurezza è il Security Configuration Manager (SCM), che consiste dei seguenti elementi:

- Modelli di protezione (Security Templates snap-in)
- Analisi e configurazione della protezione (Security Configuration and Analysis snap-in)
- Secedit.exe

LA GUI PER IL SECURITY CONFIGURATION MANAGER

Il Security Configuration Manager con interfaccia grafica consente ad un amministratore di creare e/o editare file di configurazione per la sicurezza, eseguire analisi sulla sicurezza, rivedere graficamente i risultati dell'analisi ed applicare una configurazione di sicurezza ad un sistema. Per usufruirne è necessario inserire nella Microsoft Management Console (MMC) gli

snap-in Modelli di protezione e Analisi e configurazione della protezione. In particolare per caricare i citati snap-in all'interno della MMC attivare Start-Esegui, digitare mmc nel campo Apri e all'interno della console procedere come segue:

- selezionare File-Aggiungi/Rimuovi snap-in e fare clic sul pulsante Aggiungi
- selezionare nella lista Modelli di protezione (anche noti come Security templates) e fare clic sul pulsante Aggiungi
- selezionare nella lista Analisi e configurazione della protezione e fare clic sul pulsante Aggiungi
- fare clic sul pulsante Chiudi e poi su OK

Figura 2: Aggiungere i Modelli di protezione nella Microsoft Management Console

NOTA

RIPRISTINO DEL DATABASE CORROTTO

Nel caso sia necessario recuperare un database di protezione corrotto è possibile ricorrere alla opzioni di disaster recovery di Windows. Il metodo di ripristino é il seguente: all'interno del percorso %SystemRoot%\security\Database digitare esentutl /r ma, se avete ancora problemi, eseguire esentutl con l'opzione /p all'interno della stessa cartella.

Successivamente è necessario salvare il database con le impostazioni effettuate selezionando File-Salva con nome e specificando un nome di memoria, ad esempio SCMdb.msc. Si noti che il file verrà salvato per default nella cartella Strumenti di amministrazione dell'utente attualmente loggato e sarà d'ora in poi accessibile all'interno del menu Start-Tutti i programmi-Strumenti di amministrazione.

L'Analisi e configurazione della protezione si appoggia ad un database per immagazzinare le impostazioni e le informazioni necessarie a portare a termine l'analisi stessa. A questo proposito si raccomanda che venga creato un database per ogni analisi e configurazione che si vuole portare a termine.

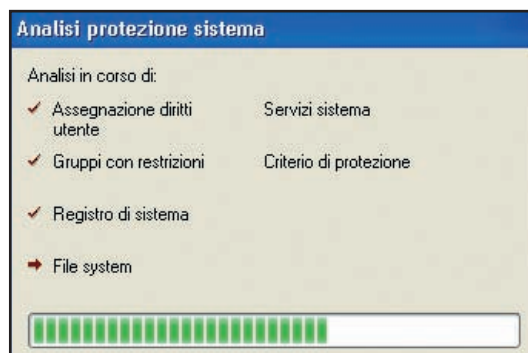



Figura 3: Eseguire l'analisi della protezione del sistema significa effettuare un confronto degli standard definiti nei security template con le impostazioni correnti del sistema, per poi procedere in maniera iterativa fino alla definizione del modello ottimale.

ANALIZZARE WINDOWS XP

Una volta eseguiti i passi precedenti è possibile effettuare un'analisi approfondita del sistema operativo per individuare eventuali falle di sicurezza. Il processo parte da un confronto degli standard definiti nei security template con le impostazioni correnti del sistema, per poi procedere in maniera iterativa fino alla definizione del modello ottimale, guidati da ciò che ci viene via via segnalato dalla Microsoft Management Console.

In particolare dovete fare un clic destro su Analisi e configurazione della protezione, selezionare Esegui analisi del sistema e premere il pulsante OK nella finestra di dialogo che identifica il file di log di riferimento (percorso C:\Documents and Settings\<utente corrente>\Documenti\Security\Logs). Al termine dell'analisi dovete entrare negli elementi all'interno di Analisi e configurazione della protezione ed esaminare nel pannello di destra gli item marcati con una croce rossa; questi ultimi, infatti, sono quelli che devono essere corretti, perché evidenziano deviazioni delle impostazioni attuali del computer rispetto agli standard definiti nel modello di protezione individuato.

In questa sede è opportuno procedere ad una vera e propria analisi degli scostamenti, per determinare in maniera certa qual è il delta e determinare il successivo piano di azione. Sebbene il modello di riferimento sia, in effetti, rigido non è affatto detto che tutti gli oggetti evidenziati in rosso portino ad azioni correttive, la cosa importante è mantenere un template come riferimento e poi definire le tolleranze che riteniamo più opportune. In pratica si dovrebbe procedere come segue:

1. Fare doppio clic su Analisi e configurazione della

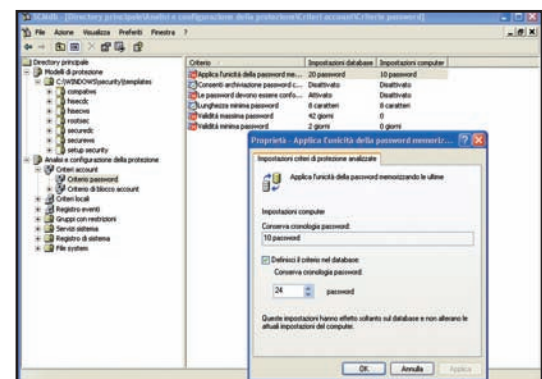
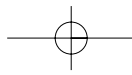



Figura 4: Gli elementi segnalati con una croce rossa identificano deviazioni delle impostazioni correnti del computer rispetto allo standard del template.

protezione e poi sugli elementi gerarchicamente inferiori, ad esempio Criteri account e, poi, Criterio password.

2. Verificare se sul pannello di destra ci siano oggetti segnalati con una croce rossa e analizzare se la deviazione delle impostazioni correnti del computer rispetto allo standard del template sono accettabili oppure no; ad esempio il criterio "Applica l'unicità delle password memorizzando le ultime", che consente di creare uno storico delle password (valori da 1 a 24) per evitare che vengano digitate sempre le stesse, può segnalare un scostamento tra le impostazioni del template (ad esempio 20) e quelle attuali del computer (ad esempio 10).
3. Ora che abbiamo i dati per procedere dobbiamo fare doppio clic sul criterio che stiamo esaminando, per poi scegliere o di escludere questo criterio dal database, togliendo il segno di spunta dalla check box "Definisci il criterio nel database", oppure d'identificare il numero delle password da conservare, accettando il valore 20 proposto dal modello o un altro valore compreso nell'intervallo di tolleranza. Se nell'esempio proposto venisse impostato, ad esempio, 10 scomparirebbe subito la croce rossa.
4. Il punto tre deve essere ripetuto per tutti gli oggetti segnalati con una croce rossa.
5. Bisogna, poi, salvare le impostazioni correnti (File-Salva o File-Salva con nome se si vuole mantenere traccia dei valori iniziali).
6. Infine fare clic destro su Analisi e configurazione della protezione e scegliere Esporta modello per creare il "nostro" modello .inf.

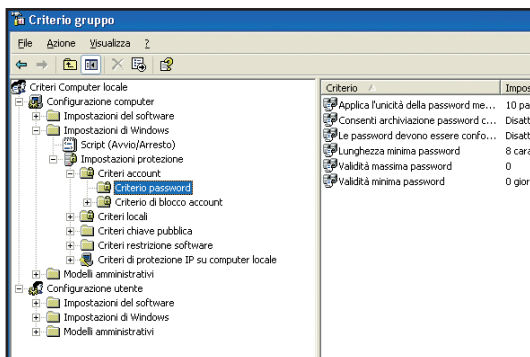



***Figura 5:** I cambiamenti eseguiti sulle policy locali vengono automaticamente inserite nel database locale delle policy e possono essere verificati direttamente nella console Criteri di gruppo.*

SIGNIFICATO DEI MODELLI DI PROTEZIONE

I Modelli di protezione sono file che contengono un set di configurazione di sicurezza e forniscono un modo semplice per standardizzare la sicurezza all'interno di un singolo sistema o di un dominio. Possono essere applicati a computer basati su Windows XP importandoli in una Group Policy oppure applicandoli direttamente sul computer locale tramite SCM. I Modelli di protezione includono un set di file predefiniti *.inf che possono essere utilizzati dal System Administrator. Il loro percorso di default è %SystemRoot%\security\templates (solitamente C:\WINDOWS\security\templates) ed hanno un significato ben preciso. In particolare Setup security.inf contiene le impostazioni predefinite di sicurezza che vengono applicate durante l'installazione del sistema operativo, compresi i permessi sui file della system root. Questo template viene utilizzato per propositi di disaster recovery e non dovrebbe mai essere applicato tramite group policy. Compatws.inf è utilizzato per modificare le autorizzazioni predefinite di accesso ai file e al registro di Windows concesse al gruppo Users in maniera tale che siano conformi ai requisiti delle principali applicazioni non certificate. Rootsec.inf è il cosiddetto modello di protezione della directory principale di sistema. Esso, infatti, definisce i nuovi permessi della system root introdotti da Windows XP Professional e puó essere utilizzato per eseguire nuovamente i permessi di accesso alla directory principale nel caso siano stati cambiati inavvertitamente.
I restanti modelli di default si possono raggruppare nelle seguenti categorie:

1. Modelli protetti, ossia i secur*.inf, che definiscono impostazioni di protezione avanzate, ad esempio per le password (essenzialmente securedc.inf e securews.inf).

2. Modelli a protezione avanzata, ossia gli hisec*.inf, che impongono ulteriori restrizioni crittografiche e di firma richieste per l'autenticazione e per i dati trasmessi sui canali protetti (essenzialmente hisecdc.inf e hisecws.inf).

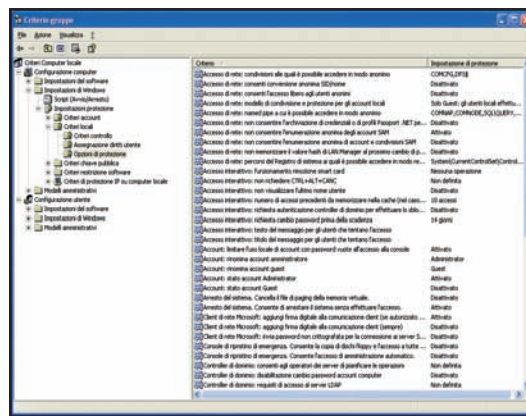
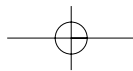



***Figura 6:** Nella console gpedit.msc, questa volta controlliamo gli aggiornamenti delle Opzioni di protezione all'interno dei Criteri locali.*

NOTA

MODIFICA E BACKUP DEL FILE .INF

I Modelli di protezione possono essere modificati a nostro piacimento; in questo modo è possibile personalizzare le policy di sicurezza che si vogliono applicare nei nostri sistemi. Tuttavia è fondamentale sottolineare che prima di effettuare qualsiasi modifica è necessario procedere al salvataggio del file .inf originario.

RENDERE EFFETTIVE LE IMPOSTAZIONI DI SICUREZZA INDIVIDUATE

L'esercizio precedente ci ha portato a creare un modello di protezione che riteniamo il più opportuno per la nostra piattaforma. Ricordiamo che fino ad ora abbiamo lavorato esclusivamente per creare il modello di riferimento, ma che non lo abbiamo ancora reso effettivo. Ora è venuto il momento di farlo. Di fatto quello che accadrá è una modifica di un certo numero di chiavi del registro di configurazione, che bisogna ovviamente salvare in anticipo.
Per applicare i setting di sicurezza dobbiamo procedere all'interno del pannello di sinistra della MMC come indicato di seguito:

1. Fare clic destro su Analisi e configurazione della protezione.
2. Selezionare dal menu a comparsa Configura il computer ora.
3. Definire il file di log di riferimento, ad esempio: %SystemRoot%\security\logs\Mysecure.log
4. Fare clic sul pulsante OK.
5. Fare poi clic su Chiudi e salvare le impostazioni della MMC, indicando il nome del file da salvare (ad es. SCMdb).
6. Eseguire il riavvio del sistema.

Una volta terminato il reboot di Windows tutte le regole definite nel security database saranno state applicate al sistema di riferimento.

VERIFICARE GLI EFFETTI

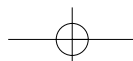
I cambiamenti eseguiti sulle policy locali vengono automaticamente inserite nel database locale delle policy. I riscontri che eseguiamo di seguito necessitano di diritti amministrativi.
Per verificare le nuove impostazioni lanciare da Start-Esegui il programma gpedit.msc, Criteri di gruppo.
Possiamo andare ora a ricercare una policy che abbiamo modificato tramite Microsoft Management Console, ad esempio quella relativa alla storicizzazione delle password. Espandiamo, quindi, Configurazione computer, Impostazioni di Windows, Impostazioni protezione ed, infine, Criteri account. Ora facciamo clic su Criterio password e a destra possiamo osservare le impostazioni che volevamo verificare. Se, poi, nel pannello di sinistra espandiamo Criteri locali e facciamo clic su Opzioni di protezione possiamo controllare a destra le ulteriori impostazioni di protezione.

LO STRUMENTO SCM PER LINEA DI COMANDO

Eseguire la Security Management Console da linea di comando significa lanciare l'utility secedit.exe che, non disponendo d'interfaccia grafica, ha evidenti limitazioni. Ovviamente è necessario avviare l'utility da Start-Esegui dopo aver digitato cmd per avviare il prompt dei comandi di Windows XP e, poi, secedit. In particolare alcune funzionalità fornite dal comando sono riportate di seguito.
L'utilità di eseguire SCM da linea di comando deve essere ricercata nella possibilitá d'inserire questo comando in uno script amministrativo per manipolare dinamicamente il sistema.
In particolare **secedit/analyze** consente di eseguire i task di analisi del sistema. L'opzione /analyze è utile ad analizzare la protezione di un sistema confrontandola con le impostazioni di base archiviate in un database. Nel comando bisogna sempre specificare il nome del database di riferimento, il nome del security template .inf e il nome del file che conterrà i risultati dell'analisi. Nel caso in cui quest'ultimo non venisse specificato é necessario consultare il log di default scesrv.log presente in %SystemRoot%\security\logs.
L'opzione **secedit /configure** consente, invece, di configurare la protezione del computer in base a quanto specificato nel database di riferimento. Anche in questo caso bisogna specificare database e modello di protezione.
secedit /export e secedit /import sono opzioni di esportazione ed importazione delle impostazioni di protezione indicate nel database. Un utilizzo pratico è la possibilità, ad esempio, d'importare determinati security template nel database corrente in maniera tale che le impostazioni definite nel modello possano essere subito applicate o, comunque, prima analizzate. Nell'esempio che segue vengono importate nel database myDB.sdb le impostazioni di protezione presenti nel file yourSec.inf.
secedit /import /db myDB.sdb /cfg yourSec.inf /overwrite
L'opzione overwrite indica che il database deve essere svuotato prima di applicare il template. Nel caso in cui /overwrite non venga specificato, le nuove impostazione andranno in append nel database ed, in caso di conflitto, i setting del nuovo modello avranno la precedenza.
Infine esiste anche la possibilità di validare la sintassi di un modello di protezione che si desidera importare nel database con la secedit /validate seguita dal nome del file. In questo caso è possibile creare un modello di ripristino con l'opzione /GenerateRollback. Tale modello potrà essere utilizzato per ripristinare i security setting presenti prima dell'applicazione del nuovo template.

NOTA

SCHEMI DI PATCHING

È importante essere a conoscenza che Microsoft utilizza i tre diversi schemi di patching (Service Pach, Hotfixes e Hotfix Rollups) non solo per i sistemi operativi, ma anche per le applicazioni. Quindi è sicuramente possibile che nonostante siano state seguite tutte le linee guide relative alla sicurezza e installate tutte le ultime patch del sistema operativo, Windows XP sia ancora vulnerabile. In altri termini un hacker potrebbe sfruttare debolezze presenti in alcune applicazioni per manomettere il sistema operativo stesso.

DISASTER RECOVERY DI DATABASE

Nel caso in cui, in seguito ad un arresto non corretto del PC, il database di protezione si sia danneggiato si può eseguire un controllo d'integrità tramite l'utility esentutl. Si tratta di posizionarsi nel percorso %SystemRoot%\security\Database e di digitare esentutl /g secedit.sdb. Una volta ultimata l'analisi verrà evidenziato un report nello stesso percorso chiamato secedit.INTEG.RAW e verificabile con Notepad o direttamente col comando type seguito dal pipe (|) e da more. Per procedere al recovery di un database danneggiato rendendolo di nuovo consistente, digitare esentutl /r nella cartella %SystemRoot%\security. Se questa operazione non ha avuto successo per ripararlo utilizzare esentutl /p nella cartella %SystemRoot%\security\Database.
Ricordarsi, infine, di cancellare i file di log creati in %SystemRoot%\security.
Nel caso siano necessarie ulteriori informazioni digitare esentutl /? per visualizzare la guida in linea con tutte le opzioni disponibili.

```
C:\WINDOWS\System32\cmd.exe

WinXP-C:\WINDOWS\security>esentutl /?

Microsoft(R) Windows(TM) Database Utilities
Version 5.1
Copyright (C) Microsoft Corporation. All Rights Reserved.

DESCRIPTION:  Maintenance utilities for Microsoft(R) Windows databases.

MODES OF OPERATION:
      Defragmentation:  ESENTUTL /d <database name> [options]
             Recovery:  ESENTUTL /r [options]
            Integrity:  ESENTUTL /g <database name> [options]
               Repair:  ESENTUTL /p <database name> [options]
             Checksum:  ESENTUTL /k <database name> [options]
            File Dump:  ESENTUTL /m[mode-modifier] <filename>

<<<<<  Press a key for more help  >>>>>
D=Defragmentation, R=Recovery, G=inteGrity, K=checKsum, P=rePair, M=File duMp
=>

RECOVERY:
    DESCRIPTION:  Performs recovery, bringing all databases to a
                  consistent state.
         SYNTAX:  ESENTUTL /r <3-character logfile base name> [options]
        OPTIONS:  zero or more of the following switches, separated by a space:
                  /l<path>    - location of log files
                                (default: current directory)
                  /s<path>    - location of system files (eg. checkpoint file)
                                (default: current directory)
                  /i          - ignore mismatched/missing database attachments
                  /d[path]    - location of database files, or current directory
                                if [path] not specified (default: directory
                                originally logged in log files)
                  /8          - set 8k database page size (default: auto-detect)
                  /o          - suppress logo

WinXP-C:\WINDOWS\security>
```

Figura 7: Le opzioni dettagliate dell'utility esentutl che consente di ripristinare un database corrotto.

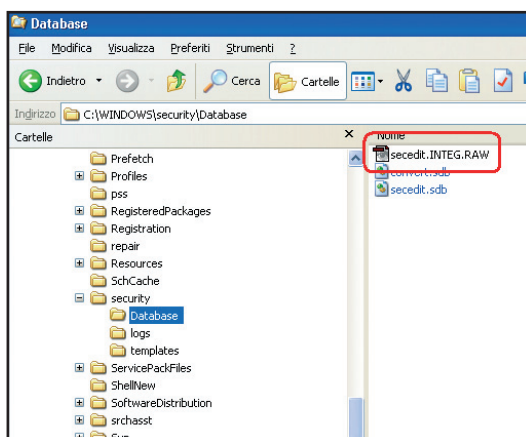



Figura 8: Dopo avere eseguito l'utility esentutl con opzione g viene generato un report chiamato secedit.INTEG.RAW

ALCUNE RACCOMANDAZIONI DA NON TRASCURARE

Al fine di fissare i buchi di sicurezza che vengono via via individuati da Microsoft è importante installare tutte le patch disponibili e, magari, abilitare sul PC l'aggiornamento automatico da Internet. A questo proposito è necessario sapere che Microsoft fornisce tre metodi di aggiornamento del sistema operativo. Il primo è quello basato sui Service Pack. Si tratta dei cosiddetti "Major Operating System updates", che includono tutte le patch più importanti e meno importanti rilasciate fino a quel momento. Sono altamente affidabili, in quanto testati accuratamente da Microsoft prima del rilascio. Il Service Pack attualmente disponibile per Windows XP è il SP2 (http://www.microsoft.com/windowsxp/sp2/default.mspx). Il secondo metodo è quello che utilizza gli Hotfixes. Tra un rilascio di un Service Pack ed un altro vengono distribuiti aggiornamenti intermedi chiamati, appunto, hotfixes. Si tratta solitamente di piccole patch che si occupano di casi specifici. Venendo rilasciati velocemente per risolvere problemi particolari, ma non sono testati accuratamente come i Service Pack. Infine esistono i cosiddetti Hotfix Rollups. Si tratta di una via di mezzo tra i due precedenti metodi di update. Non sono corposi come i Service Pack, ma contengono, in ogni caso, più di una patch, essendo dedicati alla risoluzione di più problemi. Oltre a quanto detto in precedenza è possibile aggiornare il nostro sistema operativo anche in un altro modo. È infatti possibile ottenere il supporto di Microsoft facendo il download automatico delle patch critiche di Windows XP accedendo al sito http://www.windowsupdate.com.
Non appena si è connessi al sito Web, Windows Update effettua lo scan del computer corrente per visualizzare, successivamente, un report che evidenzia le patch necessarie software e hardware. E' necessario, poi, scegliere gli update che si vogliono installare e come installarli.

Enrico Bottari

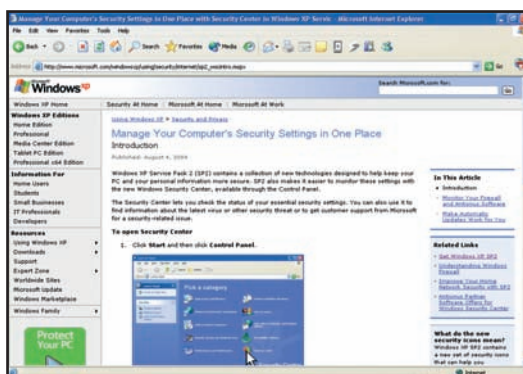
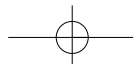



Figura 9: Windows Security Center dà una grossa mano ad identificare lo status attuale della sicurezza e ad impostare i setting di security.

CREIAMO UN VISTA PERSONALIZZATO

ALLA SCOPERTA DELLO STRUMENTO PENSATO DA MICROSOFT PER CREARE E DISTRIBUIRE IL SISTEMA OPERATIVO WINDOWS VISTA PERSONALIZZANDOLO CON LE IMPOSTAZIONI E IL SOFTWARE PIÙ ADATTO PER LE PROPRIE ESIGENZE

WAIK ovvero **Windows Automated Installation Kit**. Detto in parole semplici, WAIK è un kit composto da tool e documentazione che consente la distribuzione del sistema operativo Windows su più computer in maniera automatica. Sapere utilizzare questo strumento ci permette di accelerare notevolmente i tempi di distribuzione del software ogni qualvolta questo si rendesse necessario per un rinnovo del parco macchine o anche solo per l'applicazione di aggiornamenti e Service Pack. Dal rilascio di Windows Vista è ora disponibile un nuovo strumento: WAIK che finalmente offre a tutti la possibilità di creare un ambiente di distribuzione del nuovo sistema operativo Windows Vista. Il kit WAIK contiene oltre ad una nutrita documentazione su ciascun tool, i seguenti strumenti: Windows PE 2.0, ImageX, Windows System Image Manager, Windows Deployment Services

LA COMPOSIZIONE DEL KIT

Windows PE 2.0 è la nuova versione di Windows Preinstallation Environment, un ambiente realizzato specificatamente per consentire la preparazione di un computer nuovo per l'installazione di Windows Vista. In pratica Windows PE 2.0 è un *mini* sistema operativo (su base Windows Vista), essenziale e limitato, che permette unicamente di avviare un nuovo PC sprovvisto di sistema operativo e permetterne l'installazione attraverso un'immagine presente su disco locale o su una condivisione di rete. Chiunque abbia già installato Windows Vista, in realtà ha già conosciuto Windows PE 2.0 in quanto l'interfaccia grafica che permette l'installazione e la configurazione di Vista è proprio Windows PE 2.0 in azione.

ImageX è un tool da riga di comando che permette di creare e gestire le immagini di distribuzione di Windows con estensione WIM. In pratica questo tool permette di acquisire l'intero contenuto di una partizione del disco, comprimerla e racchiuderla all'interno di un file immagine avente estensione .WIM che potrà poi essere utilizzato in un secondo momento per ricreare la medesima partizione su un computer differente e producendo di conseguenza un clone del sistema. Questa operazione viene denominata applicazione dell'immagine ed è l'operazione speculare all'acquisizione. I comandi principali di ImageX sono:

- IMAGEX /CAPTURE
  Acquisisce l'immagine di una partizione o di una directory e la inserisce in un file WIM.
- IMAGEX /APPLY
  Applica un'immagine ad una partizione precedentemente formattata.
- IMAGEX /INFO
  Visualizza informazioni su un'immagine o su un file WIM.
- IMAGEX /SPLIT
  Divide un file WIM in più file SWM di dimensioni minori affinché questi possano essere inseriti su più supporti (CD o DVD).

***Figura 1:** ImageX nel Prompt dei comandi degli strumenti di Windows PE*

Windows System Image Manager è un tool grafico grazie al quale è possibile gestire più efficacemente e soprattutto visualmente le immagini create con ImageX. Con Windows System Image Manager possiamo modificare il contenuto dei file immagine aggiungendo o rimuovendo componenti e service pack

o modificando il file di risposte *Unattend.xml*. Questo file, molto importante, contiene in pratica tutte quelle configurazioni iniziali da applicare all'installazione del sistema operativo quali ad esempio la configurazione del disco, le impostazioni di rete, le impostazioni internazionali, e così via. Windows System Image Manager permette quindi principalmente di gestire questo file che sarebbe molto difficoltoso se non impossibile da editare a mano in quanto in formato XML ed abbastanza complesso.

I *Windows Deployment Services*, che sostituiscono i vecchi *Remote Installation Services*, sono infine un insieme di servizi che permettono di installare nuovi PC da remoto tramite rete senza la necessità di essere fisicamente presenti dinanzi ad essi. Con i Windows Deployment Services è quindi possibile ad esempio riavviare una macchina da remoto e installare il sistema operativo senza utilizzare localmente il CD o il DVD dello stesso. Questi servizi sono stati rilasciati con la Service Pack 2 di Windows Server 2003 ma sono anche disponibili in WAIK e permettono di installare uno qualsiasi dei sistemi operativi Windows, non necessariamente Windows Vista. I Windows Deployment Services sono in realtà composti da tre categorie distinte di strumenti:

- **Componenti Server**
  che comprendono un server PXE (Pre-Boot Execution Environment) e un server TFTP (Trivial File Transfer Protocol)
- **Componenti Client**
  rappresentati da una GUI che girando in Windows PE permette di comunicare con i componenti server per installare il sistema operativo.
- **Componenti di Amministrazione**
  costituiti da un insieme di tool necessari per la gestione del server, delle immagini del sistema operativo e degli account utente.

Essendo servizi server, i Windows Deployment Services possono essere installati solo su Windows Server 2003 con almeno la Service Pack 1 installata.

IL FORMATO IMMAGINE WIM

Non stiamo parlando di un nuovo formato immagine visuale, bensì del nuovo formato utilizzato per la distribuzione dei sistemi operativi Microsoft a partire dal rilascio di Windows Vista. Windows Image o WIM, ha tutta una nuova serie di caratteristiche che lo rendono realmente innovativo rispetto alle precedenti soluzioni basate su file CAB (il quale viene spesso designato come il *padre* di WIM). Innanzi tutto WIM è un formato di distribuzione indipendente dall'hardware della macchina su cui il sistema viene installato. L'unica differenza presente è quella tra sistemi a 32 o a 64 bit per i quali devono essere utilizzati due file WIM differenti. Mentre per tutte le altre configurazioni il file WIM da utilizzare resta lo stesso. Inoltre WIM può essere utilizzato per eseguire un'installazione di aggiornamento o di recovery del sistema operativo, quindi un'installazione non necessariamente distruttiva. Il team di sviluppo Microsoft ha inoltre cercato di migliorare ulteriormente la compressione delle immagini di distribuzione contenute nei file WIM arrivando ad un compromesso tra prestazioni e livello di compressione delle immagini che ora è pressoché ottimale. Ma cos'è fisicamente un file WIM? Un file WIM è un unico file che racchiude in se una o più immagini contenenti tutti i file necessari all'installazione di un sistema operativo Microsoft e comprendente:

- Un'intestazione, nella quale troviamo la firma del file WIM, la sua versione, la dimensione, un GUID (identificativo univoco) e un numero di parte, ovvero uno SplitNumber utilizzato nel momento in cui il file WIM viene fisicamente diviso su più supporti (CD o DVD).
- Dei metadati, uno per ogni immagine contenuta nel file WIM, che contengono informazioni relative ai file contenuti in ciascuna immagine quali eventuali permessi sui file, attributi, ecc.
- Una tabella risorse contenente gli HASH dei file ed utilizzata per la loro indicizzazione e rapido accesso.
- Contenuto XML vario per la descrizione del file nel suo complesso.

È da notare che un file WIM può contenere diverse immagini relative a diversi sistemi operativi. Ad esempio uno stesso file WIM potrebbe contenere l'immagine di un'installazione di Windows XP Professional e di Windows XP Home. In questo caso la novità consiste nel fatto che tutti i file uguali tra le due versioni

NOTA

DOVE TROVARE WAIK

La versione italiana di Windows Automated Installation Kit (AIK) è disponibile all'indirizzo: http://www.microsoft.com/downloads/details.aspx?familyid=C7D4BC6D-15F3-4284-9123-679830D629F2&displaylang=it

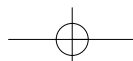
Scegliete con attenzione la lingua del kit in modo tale da farla corrispondere alla lingua della vostra versione di Windows. Se possedete Windows XP in versione inglese, scaricate WAIK in questa lingua.

NOTA

I TOOL DI LETTURA FILE ISO

I tool di lettura file ISO sono degli applicativi che permettono di leggere i file immagine con estensione .ISO, .IMG, .NRG ecc, e renderli disponibili al sistema operativo come se fossero a tutti gli effetti dei lettori CD-ROM o DVD-ROM fisici.

ATTENZIONE ALLA VERSIONE DI WINDOWS

Windows AIK può essere installato solo su Windows XP Service Pack 2, Windows Server 2003 o Windows Vista. Inoltre Windows AIK attualmente non supporta l'installazione su un sistema operativo con una lingua differente dalla propria. Per questo motivo è opportuno installare WAIK nella stessa lingua del sistema operativo in cui verrà eseguito. Se si dispone quindi di Windows in italiano, andrà installato WAIK in versione italiana.

non vengono duplicati ma dello stesso file ne troviamo una sola istanza che verrà utilizzata di volta in volta per installare l'uno o l'altro sistema operativo. In questo modo l'aumento di dimensioni dovuto alla presenza contemporanea di due installazioni nello stesso file di distribuzione corrisponderà solo alla somma delle dimensioni dei file diversi tra le due versioni di sistema operativo.

INSTALLAZIONE

Ora che sappiamo cos'è WAIK, possiamo passare alla sua installazione. Procuriamoci WAIK che viene distribuito sotto forma di file .IMG e poi con l'ausilio di un tool di lettura immagini ISO (quale ad esempio i DAEMON Tools o Virtual CD) montiamo l'immagine ed accediamo al disco di WAIK. In alternativa possiamo anche masterizzare un CD-ROM per un utilizzo più pratico. Montata l'immagine, il menu autostart di WAIK ci mostrerà una serie di utili link attraverso i quali possiamo accedere ai diversi contenuti del disco:

Figura 2: Il menu di avvio di WAIK

Clicchiamo quindi sul link *Installazione di Windows AIK* e procediamo con l'installazione che non richiede alcuna informazione particolare se non le tipiche domande dei wizard di setup (accettazione licenza, percorso di installazione, ecc). Al termine del setup troveremo nel menu Avvio il nuovo gruppo *Microsoft Windows AIK*:

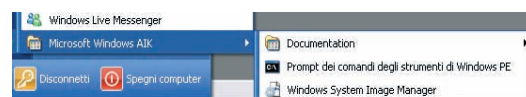



Figura 3: WAIK nel menu Avvio

Nel sotto menu *Documentation* troviamo tutta la nutrita documentazione ufficiale (fortunatamente in italiano) disponibile per WAIK che ci tornerà molto utile per comprendere i diversi passaggi implicati nella creazione di un ambiente di distribuzione funzionale.

NOTA

IMAGEX

Per informazioni dettagliate sulle opzioni della linea di comando di ImageX oltre a leggere la documentazione fornita a corredo di WAIK, possiamo anche consultare il seguente indirizzo internet: http://technet2.microsoft.com/WindowsVista/en/library/bb068119-1ba6-48c7-9ad7-3ed3f72592e91033.mspx?mfr=true

CREAZIONE DI UN AMBIENTE DI DISTRIBUZIONE

Siamo giunti quindi al momento clou della nostra conoscenza con WAIK. Ora vedremo come creare praticamente una procedura di distribuzione che ci porterà ad ottenere un'immagine di Windows Vista che potrà essere distribuita su molteplici computer. La procedura che stiamo per vedere è una simulazione e pertanto, sebbene in realtà i computer coinvolti in un processo di distribuzione reale siano diversi, nel nostro caso faremo uso di due soli computer, l'uno sul quale installeremo WAIK (il computer di lavoro) ed un secondo computer nel quale installeremo la nostra immagine del sistema operativo e che utilizzeremo anche come computer dal quale ricavare l'immagine WIM (il computer di destinazione).

Quello di cui avremo bisogno sarà quindi:

- Il DVD di Windows Vista o comunque una sua immagine ISO
- Un secondo PC, connesso in rete con il computer di lavoro
- WAIK installato sul computer di lavoro

Per prima cosa, dobbiamo creare un nuovo file di risposte, ovvero quel file (Unattend.xml) che definisce tutte le impostazioni del sistema operativo che verrà installato. Per creare un file di risposte dobbiamo avviare *Windows System Image Editor* dal gruppo di programmi Microsoft Windows AIK del menu **Avvio**.

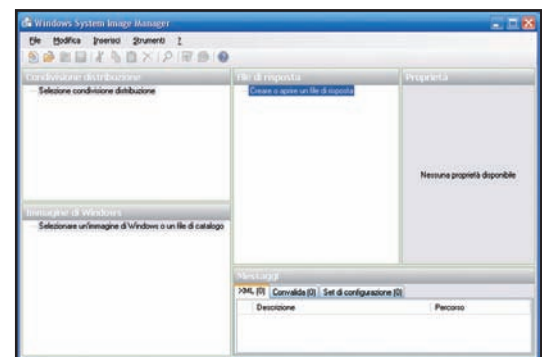
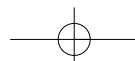
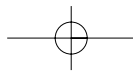



Figura 4: Windows System Image Editor

Seguiamo quindi i seguenti passaggi:

- Inserire il DVD di Windows Vista nel lettore DVD.

- Copiare il file **install.wim** dalla cartella **sources** del DVD di Vista in una cartella qualsiasi del computer di lavoro.
- In Windows System Image Editor scegliere **Seleziona immagine di Windows...** nel menu **File**.
- Selezionare quindi il file **install.wim** precedentemente copiato dal DVD di Windows Vista.
- Dalla finestra di dialogo che apparirà, selezionare l'immagine della versione di Windows Vista che si intende utilizzare e quindi premere **OK**.

A questo punto possiamo scegliere la voce **Nuovo file di risposta...** dal menu **File**. Apparirà un messaggio che ci informerà del fatto che non esiste ancora un file di catalogo:

Figura 5: Avviso di creazione nuovo file catalogo

Premiamo **OK** affinché questo file venga creato. Andiamo ora nel riquadro **Immagine di Windows** ed espandiamo il nodo **Componenti**. Qui vediamo tutti i componenti che possiamo includere / escludere dalla nostra immagine e che possiamo personalizzare tramite il file di risposte. Per un'installazione standard aggiungiamo i seguenti componenti cliccando con il tasto destro su di essi e selezionando il passaggio di configurazione specificato:

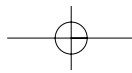
- Microsoft-Windows-Setup\DiskConfiguration\Disk\CreatePartitions\CreatePartition (Passaggio 1 windowsPE)
- Microsoft-Windows-Setup\DiskConfiguration\Disk\ModifyPartitions\ModifyPartition (Passaggio 1 windowsPE)
- Microsoft-Windows-Setup\ImageInstall\OSImage\InstallTo (Passaggio 1 windowsPE)
- Microsoft-Windows-Setup\UserData (Passaggio 1 windowsPE)
- Microsoft-Windows-International-Core-WinPE (Passaggio 1 windowsPE)
- Microsoft-Windows-Shell-Setup\OEMInformation (Passaggio 4 specialize)
- Microsoft-Windows-Shell-Setup\OOBE (Passaggio 7 oobeSystem)
- Microsoft-Windows-Shell-Setup\AutoLogon (Passaggio 7 oobeSystem)

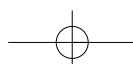
Al termine dell'aggiunta dei componenti, nel riquadro **File di risposta** dovremmo vedere i componenti appena aggiunti. Ora è necessario configurare ciascun componente impostandone le proprietà come indicato di seguito:

- Microsoft-Windows-International-Core-WinPE [**InputLocale**=it-IT, **SystemLocale**=it-IT, **UI-Language**=it-IT, **UserLocale**=it-IT]
- Microsoft-Windows-International-Core-WinPE\SetupUILanguage [**UILanguage**=it-IT]
- Microsoft-Windows-Setup\DiskConfiguration [**WillShowUI**=OnError]
- Microsoft-Windows-Setup\DiskConfiguration\Disk [**DiskID**=0, **WillWipeDisk**=true]
- Microsoft-Windows-Setup\DiskConfiguration\Disk\CreatePartitions\CreatePartition [**Extend**=false, **Order**=1, **Size**=15000, **Type**=Primary]
- Microsoft-Windows-Setup\DiskConfiguration\Disk\ModifyPartitions\ModifyPartition [**Active**=true, **Extend**=false, **Format**=NTFS, **Label**=OS_Install, **Letter**=C, **Order**=1, **PartitionID**=1]
- Microsoft-Windows-Setup\ImageInstall\OSImage [**WillShowUI**=OnError]
- Microsoft-Windows-Setup\ImageInstall\OSImage\InstallTo [**DiskID**=0, **PartitionID=1]**
- Microsoft-Windows-Setup\UserData [**AcceptEula**=true]
- Microsoft-Windows-Setup\UserData\ProductKey [**Key**=<Windows Product Key>, **WillShowUI**=OnError]
- Microsoft-Windows-Shell-Setup\OEMInformation [**Manufacturer**=<produttore>, **HelpCustomized**=false]
- Microsoft-Windows-Shell-Setup\OOBE [**HideEULAPage**=true, **ProtectYourPC**=3, **SkipMachineOOBE**=true, **SkipUserOOBE**=true, **NetworkLocation**=Work]
- Microsoft-Windows-Shell-Setup\AutoLogon [**Enabled**=true, **LogonCount**=5, **Username**=Administrator]
- Microsoft-Windows-Shell-Setup\AutoLogon\Password [<password amministratore>]

Terminata l'aggiunta e la configurazione dei componenti, possiamo finalmente validare e salvare il nostro nuovo file di risposte, selezionando prima la voce **Convalida file di risposta** dal menu Strumenti e poi, in caso di esito positivo, **Salva file di risposta** dal menu **File** dando come nome al nostro file di risposte **Autounattend.xml** e posizionando lo stesso su una chiave di memoria USB (vedremo in seguito il motivo di questa operazione).

I WINDOWS DEPLOYMENT SERVICES

I Windows Deployment Services non sono installati automaticamente insieme a WAIK ma devono essere installati esplicitamente eseguendo il file windows-deployment-services-update-x86.exe o il file windows-deployment-services-update-amd64.exe (in base alla tipologia del proprio processore) presenti nel percorso <root>:\WDS del CD di WAIK.

WEB REFERENCE

Informazioni generali sulle tecnologie di deployment di Microsoft:
http://www.microsoft.com/technet/desktopdeployment/default.mspx

GIANNI MALANGA
Lavora da più di 8 anni con tecnologie Microsoft in particolare nel campo dello sviluppo Web. Sviluppa in .NET ormai da alcuni anni e ha svolto diverse docenze per corsi di formazione professionali. Dopo aver lavorato alle dipendenze di importanti realtà locali, attualmente ha intrapreso la libera professione costituendo la ditta individuale Kaone Consulting che si occupa di consulenza informatica per PMI su tutto il territorio nazionale. Gli si può scrivere all'indirizzo kaone@email.it e il suo blog è disponibile su: http://thekaone.blogspot.com

Siamo arrivati quindi all'installazione di Windows sul secondo PC, quello di destinazione. Inseriamo la chiave di memoria USB in una porta USB libera del computer ed installiamo Windows Vista dal suo DVD. Per impostazione predefinita, il setup di Windows Vista cerca in tutte le unità rimovibili un file *Autounattend.xml* avendo posizionato il nostro file *Autounattend.xml* sulla chiave di memoria USB, l'installazione di Windows Vista lo troverà e lo utilizzerà per avviare l'installazione sulla base delle impostazioni da noi effettuate con Windows System Image Editor. Al termine dell'installazione del sistema operativo dobbiamo preparare il computer alla generazione dell'immagine di distribuzione e quindi per fare questo eseguiamo dal prompt dei comandi Sysprep:

```
c:\windows\system32\sysprep\sysprep.exe /oobe
                              /generalize /shutdown
```

Al termine della preparazione, Sysprep effettuerà lo shutdown del sistema che ora è pronto per essere acquisito come immagine. Per acquisire un'immagine del sistema con ImageX, utilizziamo l'ambiente Windows PE. Dobbiamo quindi creare un disco di avvio di Windows PE 2.0 seguendo questi semplici passaggi sul computer di lavoro:

- Eseguire al prompt dei comandi degli strumenti di Windows PE:
  ```
  copype x86 c:\winpe
  ```
- Copiare imagex:
  ```
  copy "c:\programmi\Windows
           AIK\Tools\x86\imagex.exe" c:\winpe\iso\
  ```
- Creare un file di immagine ISO di Windows PE con il comando:
  ```
  oscdimg -n -bc:\winpe\etfsboot.com c:\winpe\ISO
                                  c:\winpe\winpe.iso
  ```

Possiamo ora masterizzare un CD-ROM con l'immagine ISO di Windows PE ed utilizzarlo per avviare il computer di destinazione affinché parta dal CD di Windows PE (nel caso il computer non fosse configurato per partire da CD-ROM, modificare opportunamente la sequenza di avvio nel BIOS della macchina). Al termine della procedura di avvio vedremo apparire la console a riga di comando di Windows PE.

Figura 6: Console di comando di Windows PE 2.0

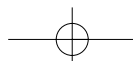
Procediamo quindi con l'acquisizione dell'immagine del computer di destinazione digitando al prompt dei comandi di Windows PE il seguente comando:

```
d:\imagex /compress fast /capture c: c:\winimg.wim
                   "Windows Vista Installation" /verify
```

Ottenuta l'immagine **winimg.wim** possiamo ora comodamente copiarla sul nostro computer di lavoro in una condivisione di rete in modo tale da renderla disponibile per l'applicazione su altri computer.

LA DISTRIBUZIONE

Possiamo ora a scopo di test formattare totalmente il computer di destinazione che abbiamo utilizzato per generare l'immagine di distribuzione. Effettuata la formattazione, inseriamo nel computer il CDROM di Windows PE generato precedentemente e avviamo il computer. Quando verrà mostrata la console di Windows PE possiamo copiare l'immagine **winimg.wim** dalla condivisione di rete del nostro computer di lavoro, sul computer di destinazione appena formattato:

```
copy \\mypc\immagine\winimg.wim c:\
```

e poi eseguire nuovamente ImageX, questa volta per applicare l'immagine anziché generarla:

```
d:\imagex /apply c:\winimg.wim l c:
```

Dove 1 indica l'immagine da applicare che nel nostro caso è la prima, visto che il file WIM da noi generato contiene solo un'immagine. Queste ultime operazioni possono essere eseguite su tutti i computer che si intende installare a partire dall'immagine creata e quindi in un numero indefinito di volte che è proprio lo scopo principale di WAIK.

CONCLUSIONI

Microsoft con il rilascio di WAIK ha voluto semplificare notevolmente la distribuzione e l'upgrade dei computer verso il suo nuovo sistema operativo Windows Vista ed anche per questo ha deciso di distribuire gratuitamente a tutti questo kit affinché chiunque potesse utilizzarlo per avviare il processo di migrazione nella propria azienda o nell'azienda per cui lavora. Certo WAIK non è uno strumento alla portata di chiunque in quanto è comunque indispensabile una certa esperienza nell'utilizzo di questo genere di applicazioni ma è sempre in ogni caso molto più amichevole rispetto agli strumenti del passato.

APPLICAZIONI PRATICHE DI XAML

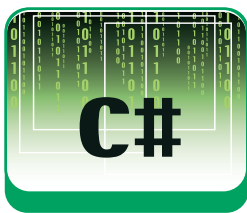
XAML È LA NOVITÀ DEL MOMENTO E MOLTO SE NE STA PARLANDO MA COSA POSSIAMO REALMENTE FARE CON QUESTA NUOVA TECNOLOGIA? VEDIAMOLO CON UN ESEMPIO CONCRETO. IMPAREREMO COME SFRUTTARE IL DRAG & DROP

XAML è la moda del momento così come lo è stato e lo è ancora Ajax. A differenza di quest'ultimo, però, XAML insieme a tutto il nuovo .NET Framework 3.0 è realmente una novità, in quanto prima del loro arrivo non era possibile sviluppare applicazioni così ricche graficamente se non con notevoli sforzi e con l'utilizzo di tecnologie molto complesse. Uno degli aspetti più visibili ed accattivanti di questa nuova tecnologia è rappresentato dalle animazioni e dai nuovi oggetti panel dedicati alla visualizzazione degli elementi grafici. È su questi infatti che ci concentreremo maggiormente. Nel presente articolo quindi affronteremo lo sviluppo in ambiente web e vedremo come realizzare in WPF/E, ovvero Windows Presentation Foundation Everywhere, il famosissimo gioco del "Quindici" a cui i meno giovani di noi sicuramente ricorderanno di aver giocato nella loro infanzia. Spieghiamo brevemente in cosa consiste questo gioco per coloro che non lo conoscessero. In pratica il gioco del Quindici nella sua versione tradizionale, era un quadratino in plastica contenente quindici tesserine numerate da 1 a 15 libere di muoversi di una posizione in verticale od in orizzontale. L'intera base contenente le caselline aveva una dimensione di 4x4 e quindi si veniva a creare uno spazio vuoto. Lo spazio in più doveva essere sfruttato per muovere le quindici tesserine di una posizione per volta. Scopo del gioco era quello di riordinare tutte le tesserine da uno a quindici partendo naturalmente da uno stato disordinato. Vediamo nella seguente figura il risultato che otterremo al termine dell'articolo:

Figura 1: Il gioco del quindici in XAML all'avvio

LA LOGICA DI GIOCO

Uno dei concetti base di XAML è rappresentato dagli elementi contenitori o Panel ovvero un contenitore di oggetti come lo era il Panel delle versioni precedenti del .NET Framework. Ma a differenza di prima ora XAML ci mette a disposizione diversi tipi di Panel ognuno con una propria specializzazione. Tra i vari tipi di panel troviamo l'oggetto Grid che per la sua natura è quello che si presta maggiormente alla creazione del nostro gioco. In pratica l'oggetto Grid è un panel che suddivide il suo contenuto per righe e colonne come ad esempio avviene per una tabella HTML:

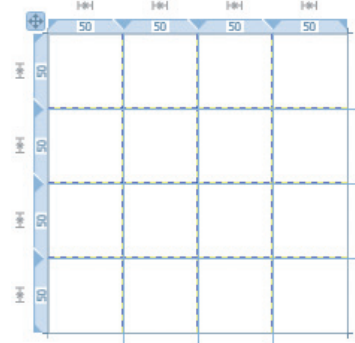



Figura 2: L'oggetto Grid in Visual Studio

Ogni oggetto contenuto in un Grid eredita da questo due proprietà (Grid.Column e Grid.Row) che indicano rispettivamente la Colonna e la Riga in cui l'oggetto è posizionato e attraverso la modifica di queste proprietà è possibile spostare l'oggetto in una qualsiasi delle celle della griglia. Sostanzialmente ciò che faremo sarà creare un Grid di 4 colonne per 4 righe in cui posizioneremo quindici Label a rappresentare le nostre tessere del gioco lasciandone una vuota che sarà la nostra tessera "zero". Ciascuna Label visualizzerà un numero e potrà essere spostata in orizzontale od in verticale nel caso in cui questa sia adiacente alla cella libera. Dovremo quindi gestire l'evento *MouseDown* dell'oggetto Grid e verificare se la cella su cui si è cliccato confina con la cella libera. In tal caso modifi-

cheremo opportunamente le proprietà Column e Row per spostare la Label nella cella prima occupata dalla casella vuota. Per abbellire un po' il tutto applicheremo un semplice effetto di sfumatura alla cella durante lo spostamento, inseriremo un contatore di spostamenti per capire, in caso di vittoria, in quante mosse siamo riusciti a risolvere il gioco ed infine visualizzeremo anche un piccolo timer che ci dirà il tempo trascorso dall'inizio del gioco.
Per quanto riguarda il timer, se proviamo ad utilizzare il tipico oggetto Timer del namespace *System.Timers* ci accorgiamo subito che non è possibile usarlo in quanto alla prima esecuzione e al verificarsi del primo evento *Elapsed* verrà sollevata l'eccezione:

```
System.InvalidOperationException: The calling thread
  cannot access this object because a different thread
                                                owns it
```

Questo accade perché gli oggetti dell'interfaccia grafica di WPF vengono instanziati in un thread diverso da quello in cui vengono creati i timer del namespace *System.Timers*. Non potendo utilizzare il classico Timer a cui siamo abituati, ci viene in aiuto un nuovo oggetto timer chiamato *DispactherTimer* che a differenza del primo, gira nello stesso thread degli oggetti di interfaccia di WPF e quindi può accedere ad essi.
Dopo aver installato le estensioni *Visual Studio 2005 extensions for .NET Framewok 3.0 (WCF & WPF)* troviamo tra i project template di Visual Studio nel ramo *NET Framework 3.0* il nuovo template **XAML Browser Application (WPF)**. Creato quindi un nuovo progetto XAML Browser troviamo nel progetto alcuni file tra cui quello che ci interessa è *Page1.xaml* ovvero la nostra pagina XAML nella quale andremo ad inserire tutti gli oggetti dell'interfaccia del nostro gioco. Nel codice allegato trovate l'intero progetto che ora andremo a commentare.

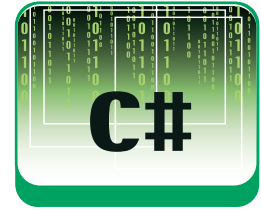
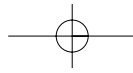
IMPLEMENTAZIONE

Nella pagina XAML è presente un pannello Canvas che imposta le dimensioni massime della griglia e che contiene la Label visualizzata in caso di vittoria. All'interno di questo Canvas troviamo uno *StackPanel*, ovvero un pannello il cui contenuto viene "impilato" in modo tale da essere visualizzato ordinatamente in verticale od in orizzontale. Nel nostro caso l'ordinamento sarà verticale. All'interno dello StackPanel troviamo l'oggetto *Grid*, subito sotto troviamo il Button che permette di ricominciare il gioco e mescolare le tessere ed infine troviamo un *WrapPanel* che ospita diverse Label, tra cui il contatore degli spostamenti ed il tempo trascorso dall'inizio del gioco.
Vediamo ora più nel dettaglio la composizione dell'oggetto Grid. L'oggetto Grid prevede la definizione di due sezioni *Definitions*, ovvero della definizione delle colonne e delle righe della griglia:

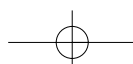
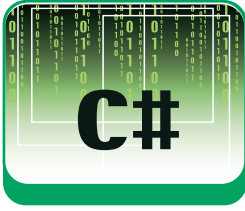
```
<Grid.ColumnDefinitions>
        <ColumnDefinition Width="50" />
        <ColumnDefinition Width="50" />
        <ColumnDefinition Width="50" />
        <ColumnDefinition Width="50" />
</Grid.ColumnDefinitions>

<Grid.RowDefinitions>
        <RowDefinition Height="50" />
        <RowDefinition Height="50" />
        <RowDefinition Height="50" />
        <RowDefinition Height="50" />
</Grid.RowDefinitions>
```

Queste definizioni specificano di creare all'interno del Grid quattro colonne e quattro righe di dimensione 50 **DIP**. Cogliamo l'occasione per specificare che i valori utilizzati per le proprietà Width ed Height non sono, come si potrebbe credere, specificati in semplici pixel, bensì in una nuova unità di misura denominata Device Independent Pixels, ovvero DIP. Un DIP equivale ad 1/96 di pollice e questa unità di misura è stata utilizzata per permettere alle applicazioni di poter adattarsi alla risoluzione dello schermo su cui vengono visualizzate e quindi poter essere viste senza problemi su device molto diversi tra loro (quali ad esempio palmari, telefoni cellulari, PDA, ecc).
Subito dopo definiamo tutte le Label che saranno contenute nella griglia e che visualizzeranno i numeri delle tessere del gioco:

```
<Label Grid.Column="0" Grid.Row="0" Width="48"
            Height="48" Name="lblUno" FontSize="22"
                HorizontalContentAlignment="Center"
                          Background="LightBlue"
        VerticalContentAlignment="Center">1</Label>

<Label Grid.Column="1" Grid.Row="0" Width="48"
            Height="48" Name="lblDue" FontSize="22"
                HorizontalContentAlignment="Center"
                          Background="LightBlue"
        VerticalContentAlignment="Center">2</Label>
…..
…..
…..
```

Ogni Label imposta le proprietà Column e Row ereditate dall'elemento Grid. In ogni caso questa prima impostazione di partenza con le Label ordinate verrà successivamente variata quando le caselle saranno mescolate in modo random.

NOTA

REFERENZIARE DLL NEI PROGETTI

In Visual Studio per referenziare un assembly in un progetto è sufficiente cliccare con il tasto destro del mouse sul nome del progetto, scegliere la voce Add Reference… , cliccare sul tab Browse e selezionare dal disco rigido la DLL che s'intende referenziare nel progetto.

Subito sotto la griglia troviamo il tasto per l'inizializzazione del gioco:

```
<Button Name="cmdNewGame" Width="100"
        Padding="0,0,2,1" Margin="0,10,0,0">Nuovo
                                    Gioco</Button>
```

In questo caso non troviamo alcuna specificità se non la presenza delle proprietà *Padding* e *Margin* che si riferiscono rispettivamente allo spazio tra il bordo del tasto ed il suo contenuto interno e la distanza tra i bordi del tasto e gli elementi che lo circondano.
Per ultimo troviamo il WrapPanel che contiene le Label degli spostamenti e del tempo impiegato:

```
<WrapPanel HorizontalAlignment="Center"
                              Margin="0,10,0,0">
<Label Width="110" BorderBrush="Black"
              BorderThickness="1" Height="24"
       Name="lblCounter" Background="#44808080"
     FontWeight="Bold" HorizontalAlignment="Center"
          HorizontalContentAlignment="Center">0
                              spostamenti</Label>
<Label Width="94" Height="24"
     Name="lblTimerTitle" Background="#44008080"
     FontFamily="Bold" HorizontalAlignment="Center"
       HorizontalContentAlignment="Right">Tempo
                                trascorso: </Label>
<Label Height="24" Name="lblTimer"
     Background="#44008080" FontFamily="Bold"
                    HorizontalAlignment="Stretch"
HorizontalContentAlignment="Left">00:00:00</Label
                                                >
<Label Height="24" Name="lblTimerSec"
     Background="#44008080" FontFamily="Bold"
                    HorizontalAlignment="Stretch"
     HorizontalContentAlignment="Left">sec.</Label>
</WrapPanel>
```

All'interno del tag Page troviamo la sezione Page.Resources:

```
<Page.Resources>
  <DoubleAnimation
      x:Key="swapAnimation"
      Duration="0:0:1.0" />
  <Storyboard
      x:Key="noSwapStoryBoard"
Storyboard.TargetProperty="(Background).(SolidColor
                                      Brush.Color)">
      <ColorAnimation
  From ="LightBlue"
  To ="Red"
  Duration="0:0:0.1"
  AutoReverse="True" />
  </Storyboard>
</Page.Resources>
```

Questa sezione definisce le risorse condivise di pagina. Possiamo considerare le risorse di pagina come una *Hashtable* che può contenere una serie di oggetti a cui possiamo accedere specificando una chiave. In questo caso la nostra chiave è rappresentata dall'attributo *x:Key="chiave"*.
Nelle risorse abbiamo inserito una *DoubleAnimation* che usiamo per animare una Label nel momento in cui questa viene spostata ed uno *Storyboard* che utilizziamo per animare una Label che non può essere spostata.
Nel primo caso la definizione completa dell'animazione è presente nel code behind della pagina perché, come vedremo, abbiamo la necessità di riutilizzare l'animazione con differenti parametri, mentre per quanto riguarda l'animazione dello Storyboard, questa è definita direttamente in XAML in quanto non prevediamo di modificarne da codice i parametri.
Passiamo ora al code behind della pagina XAML. Nel costruttore della classe Page, inizializziamo la griglia disponendo casualmente le tessere e registriamo i gestori di evento del click del mouse sul tasto per la creazione di un nuovo gioco e sulla griglia per la gestione del clic sulle caselle.

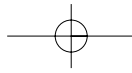
```
public Page1()
{
    InitializeComponent();
    InitializeGridCells();
    Grid.MouseDown += new
        MouseButtonEventHandler(Grid_MouseDown);
    cmdNewGame.Click += new
            RoutedEventHandler(cmdNewGame_Click);
```

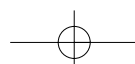
Per quanto riguarda la gestione degli eventi, in XAML abbiamo due possibilità. Possiamo definire i gestori di evento direttamente in XAML:

```
<Button Name="cmdNewGame"
 Click="cmdNewGame_Click">Nuovo Gioco</Button>
```

Oppure possiamo associare gli Event Handler da code behind, come abbiamo fatto nel nostro caso:

```
cmdNewGame.Click += new
            RoutedEventHandler(cmdNewGame_Click);
```

Al momento l'unica differenza apprezzabile nella scrittura, tra queste due metodologie, è riscontrabile nel fatto che nel primo caso l'event handler ci viene creato direttamente da Visual Studio premendo due volte il tasto TAB subito dopo aver digitato i caratteri += mentre nel secondo caso dobbiamo scrivere noi manualmente il metodo del gestore. Ma questa è evidentemente una limitazio-

IL .NET FRAMEWORK 3.0

Per eseguire il codice allegato a questo articolo e per sviluppare in XAML è necessario avere già installato sulla propria macchina il .NET Framework 3.0 che è possibile recuperare a questo indirizzo: http://www.microsoft.com/downloads/details.aspx?FamilyId=10CC340B-F857-4A14-83F5-25634C3BF043&displaylang=en

ne del Designer di Visual Studio che sarà presto superata.
Creiamo inoltre il timer che ci servirà per visualizzare il tempo trascorso dall'inizio del gioco:

```
    _timer = new DispatcherTimer();
    _timer.Interval = new TimeSpan(0, 0, 1);
    _timer.Tick += new EventHandler(_timer_Tick);
    _timer.Start();
}
```

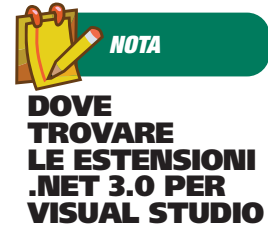
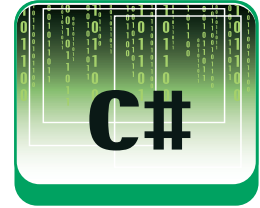
Il metodo preposto all'inizializzazione della griglia utilizzando la classe Random, genera 16 coppie di numeri casuali che diventeranno le righe e le colonne delle sedici caselle del gioco. In questo metodo vediamo la collection *Children* dell'oggetto Grid che espone tutti gli oggetti (di tipo *UIElement*) presenti nella griglia, nel nostro caso i children sono le Label contenenti i numeri delle tessere del gioco. Attraverso l'utilizzo dei due metodi *Grid.SetColumn* e *Grid.SetRow* è possibile impostare la riga e la colonna in cui vogliamo posizionare uno specifico elemento dell'interfaccia:

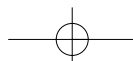
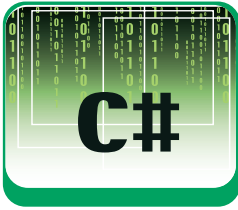
```
private void InitializeGridCells()
{
    Random rnd = new
                Random(DateTime.Now.Millisecond);
    List<string> colsAndRows = new
                List<string>(16);
    int col = 0;
    int row = 0;
    while (colsAndRows.Count < 16)
    {
      col = rnd.Next(0, 4);
      row = rnd.Next(0, 4);
      if (!colsAndRows.Contains(col.ToString() + "|"
                                + row.ToString()))
      {
        colsAndRows.Add(col.ToString() + "|" +
                                row.ToString());
      }
    }
    int i = 0;
    foreach (UIElement elem in Grid.Children)
    {
      Grid.SetColumn(elem,
            int.Parse(colsAndRows[i].Split('|')[0]));
      Grid.SetRow(elem,
            int.Parse(colsAndRows[i].Split('|')[1]));
      i++;
    }
}
```

Il gestore dell'evento MouseDown del Grid verifica che il click sia stato effettuato su di una Label, esegue lo spostamento della casella (se questo è consentito) e quindi avvia l'animazione che visualizzerà un effetto di sfumatura sulla Label cliccata. Se lo spostamento non è consentito, viene avviato lo Storyboard ritornato dal metodo GetErrorStoryBoard e dedicato alla visualizzazione di un'animazione d'errore:

```
private void Grid_MouseDown(object sender,
                    MouseButtonEventArgs e)
{
  object obj = e.Source;
  if (obj is Label)
  {
      Label currentLabel = ((Label)obj);

      if (SwapCell((UIElement)obj))
      {
currentLabel.BeginAnimation(Label.OpacityProperty,
              GetAnimation(0.0, 1.0, 0.1, false));
      }
      else
      {
currentLabel.BeginStoryboard(GetErrorStoryBoard(
                          currentLabel.Name));
      }
  }
}
```

Il metodo *BeginAnimation* richiede come parametri la proprietà dell'oggetto che vogliamo modificare (e quindi animare) e *un'AnimationTimeline*, cioè un oggetto che definisce il tipo di valore da assegnare alla proprietà da animare ed il tempo che deve trascorrere tra l'inizio ed il termine dell'animazione. Nel nostro caso utilizziamo una *DoubleAnimation* (restituitaci dal nostro metodo *GetAnimation*) che farà variare la proprietà *OpacityProperty* dal valore 0.0 al valore 1.0 a passi di 0.1 nell'arco di tempo di un secondo. Il metodo *BeginStoryboard* richiede invece uno Storyboard da associare all'oggetto specificato. Anche in questo caso abbiamo un metodo che restituisce lo Storyboard dato il nome dell'oggetto a cui intendiamo applicarlo.
Con *GetAnimation* ritorniamo un oggetto DoubleAnimation che rappresenta un'animazione, ricevendo come parametri il valore di partenza, il valore di arrivo, il valore dello step ed il valore della proprietà *AutoReverse* che indica se ripetere l'animazione quando questa è arrivata al valore di arrivo oppure fermarsi. L'oggetto DoubleAnimation viene recuperato dalle risorse tramite l'uso del metodo *FindResource*:

```
private DoubleAnimation GetAnimation(double from,
          double to, double by, bool autoReverse)
{
    _da =
  (DoubleAnimation)FindResource("swapAnimation");
```

DOVE TROVARE LE ESTENSIONI .NET 3.0 PER VISUAL STUDIO

Le estensioni necessarie per permettere di sviluppare in .NET 3.0 con Visual Studio, possono essere scaricate dal seguente indirizzo:
http://www.microsoft.com/downloads/details.aspx?familyid=F54F5537-CC86-4BF5-AE44-F5A1E805680D&displaylang=en

```
        // Animazione casella spostata
    _da.From = from;
    _da.To = to;
    _da.By = by;
    _da.AutoReverse = autoReverse;

    return _da;
}
```

Invece il metodo *GetErrorStoryBoard* ritorna un oggetto Storyboard (anch'esso recuperato dalle risorse di pagina) dopo aver impostato la sua proprietà TargetName che gli indicherà la Label sulla quale applicare l'animazione:

```
private Storyboard GetErrorStoryBoard( string
                                        targetName)
{
    _errStoryBoard =
    (Storyboard)FindResource("noSwapStoryBoard");
_errStoryBoard.SetValue(Storyboard.TargetNameProp
                                erty, targetName);
    return _errStoryBoard;
}
```

A questo punto vediamo il metodo *SwapCell* che effettua materialmente lo spostamento delle celle. Innanzitutto ricaviamo la riga e la colonna della Label vuota e della Label che intendiamo spostare utilizzando i metodi *Grid.GetColumn* e *Grid.GetRow* dell'oggetto Grid:

```
private bool SwapCell(UIElement swappedCell)
{
    bool retVal = true;

    // Riga e Colonna della cella Zero
    int zColumn = Grid.GetColumn(lblZero);
    int zRow = Grid.GetRow(lblZero);
    // Riga e Colonna della cella selezionata
    int tmpColumn = Grid.GetColumn(swappedCell);
    int tmpRow = Grid.GetRow(swappedCell);
```

Poi con un semplice calcolo verifichiamo che la Label da spostare confini in orizzontale od in verticale con la Label vuota:

```
    if ((zColumn == tmpColumn && Math.Abs(zRow
                                  - tmpRow) == 1)

      | (zRow == tmpRow && Math.Abs(zColumn -
                               tmpColumn) == 1))
    {
```

Due celle sono contigue semplicemente se sono sulla stessa colonna e distano tra loro di una riga oppure se sono sulla stessa riga e distano tra loro di una colonna.

NOTA

WINFX SDK

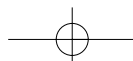
L'SDK per sviluppare in XAML così come altro utile materiale per sviluppare in .NET 3.0 lo potete trovare a questo indirizzo: http://msdn.microsoft.com/winfx/downloads/

Verificato quindi con questo calcolo che la Label cliccata può essere spostata, utilizzando i metodi *Grid.SetColumn* e *Grid.SetRow* invertiamo le posizioni delle due Label ottenendo come risultato che queste si scambieranno di posto:

```
        Grid.SetColumn(swappedCell, zColumn);
        Grid.SetRow(swappedCell, zRow);
        Grid.SetColumn(lblZero, tmpColumn);
        Grid.SetRow(lblZero, tmpRow);
        UpdateCounter( false);
    }
    else
    {
        retVal = false;
    }
```

Subito dopo lo swap aggiorniamo il contatore degli spostamenti con il metodo *UpdateCounter*.
Ora possiamo controllare che lo schema sia stato completato, verificando con il nostro metodo *IsCompleted* che tutte le celle siano nella posizione corretta. Se così è, visualizziamo la Label *lblWin* che mostra il messaggio di vittoria applicando anche ad essa un'animazione che le darà l'effetto di partire dal centro dello schermo ed ingrandirsi fino a raggiungere le dimensioni dell'intera griglia:

```
    if (IsCompleted())
    {
        lblWin.Visibility = Visibility.Visible;
        lblWin.BeginAnimation(Label.WidthProperty,
                     GetAnimation( 0, 200, 10, false));

        lblWin.BeginAnimation(Label.HeightProperty,
                      GetAnimation(0, 200, 10, false));
        lblWin.BeginAnimation(Canvas.TopProperty,
                      GetAnimation(100, 0, 10, false));

        lblWin.BeginAnimation(Canvas.LeftProperty,
                      GetAnimation(100, 0, 10, false));
        _timer.Stop();
    }
    return retVal;
}
```

Anche in questo caso abbiamo utilizzato il metodo *BeginAnimation* della Label passandogli il nome della proprietà che vogliamo modificare ed il tipo di animazione da applicare. Nel caso specifico animeremo contemporaneamente le quattro proprietà *Label.WidthProperty*, *Label.HeightProperty*, *Canvas.TopProperty* e *Canvas.LeftProperty*. Quindi al termine dell'animazione, fermiamo il timer che lascerà visibile il tempo impiegato per risolvere il gioco.

AVVIAMO IL GIOCO

Per giocare con il nostro gioco del Quindici, premiamo F5 ed attendiamo che Visual Studio faccia partire Internet Explorer. A questo punto possiamo iniziare a giocare spostando le caselle cliccandole con il mouse. Ogni spostamento farà incrementare il contatore e nel frattempo il timer visualizzerà il tempo trascorso dall'inizio del gioco. Quando riusciremo a ordinare tutte le celle da uno a quindici, vedremo apparire con un'animazione la Label di vittoria.

Figura 3: *Label visualizzata in caso di vittoria.*

A questo punto possiamo premere il tasto *New Game* per rimescolare le tessere ed iniziare una nuova partita. Oltre che lanciare l'applicazione da Visual Studio possiamo però anche pubblicarla su IIS come una qualsiasi Web Application. Per fare questo è sufficiente:

1. cliccare con il tasto destro del mouse sul nome del progetto
2. selezionare *Publish*...
3. selezionare la virtual directory in cui si intende pubblicare l'applicazione
4. selezionare l'opzione per rendere disponibile l'applicazione solo in modalità online
5. premere *Finish* per confermare

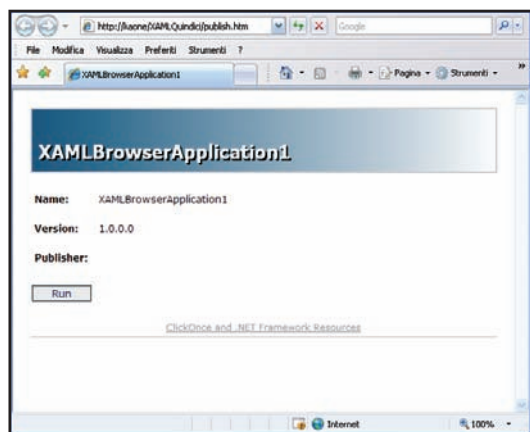



Figura 4: *Avvio installazione ClickOnce dell'applicazione XBAP*

Al termine della pubblicazione, verrà avviato Internet Explorer con caricata la pagina publish.htm la quale ci richiederà l'installazione dell'applicazione (ricordiamo che le applicazioni WPF/E vengono installate tramite *ClickOnce*). Premiamo quindi il tasto *Run*:

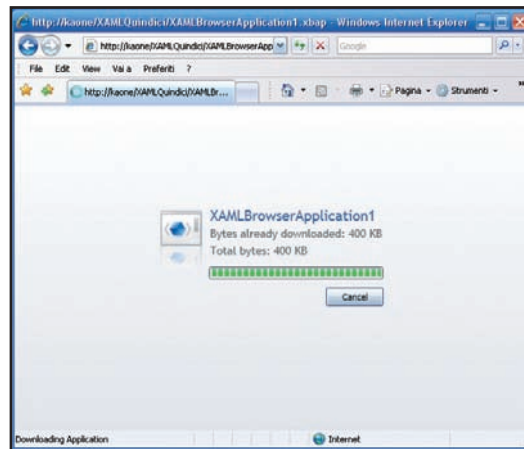
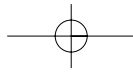
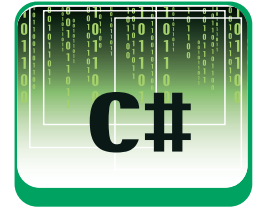



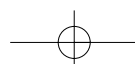
Figura 5: *Installazione dell'applicazione XBAP*

Alla pressione del tasto Run verrà avviata l'installazione dell'applicazione:

Al termine dell'installazione finalmente vedremo apparire il nostro gioco del Quindici caricato da IIS e quindi pubblicabile anche online. Noteremo subito che il file caricato da Internet Explorer non avrà estensione html, xaml o aspx, bensì XBAP (XAML Browser Application). Questa è l'estensione delle applicazioni scritte in XAML per WPF/E e che vengono interpretate dall'applicazione PresentationHost.exe (situata in C:\Windows\system32) preposta all'interpretazione del codice scritto in WPF per il web.

CONCLUSIONI

In questo articolo abbiamo visto solo una piccola parte di tutto quello che XAML permette di fare ma è sufficiente per introdurci gradualmente ai nuovi concetti che tra breve dovremo imparare a padroneggiare per essere al passo con i tempi. Molte cose al momento in cui scriviamo sono allo stato grezzo ed ancora in fase di sviluppo e quindi alcune cose che vanno fatte manualmente possono sembrare complicate ma quando tra breve queste tecnologie saranno mature e verranno rilasciate le versioni definitive dei diversi componenti e tool di sviluppo, il tutto sarà molto più facile da implementare e da utilizzare e finalmente potremo fornire applicazioni dall'aspetto più innovativo e user friendly.

Gianni Malanga

GIANNI MALANGA

Lavora da più di 8 anni con tecnologie Microsoft in particolare nel campo dello sviluppo Web. Sviluppa in .NET ormai da alcuni anni e ha svolto diverse docenze per corsi di formazione professionali. Dopo aver lavorato alle dipendenze di importanti realtà locali, attualmente ha intrapreso la libera professione costituendo la ditta individuale Kaone Consulting che si occupa di consulenza informatica per PMI su tutto il territorio nazionale. Gli si può scrivere all'indirizzo kaone@email.it

DISEGNAMO UN FORM CON WPF

RIDEFINIRE UN FORM CON .NET NON È COSA IMPOSSIBILE MA TUTT'ALTRO CHE BANALE E NON SEMPRE CON RISULTATI ENTUSIASMANTI. OGGI, GRAZIE AGLI OGGETTI MESSI A DISPOSIZIONE DAL NUOVO ENGINE GRAFICO WPF, TUTTO È SEMPLIFICATO

Conoscenze richieste
Conoscenze minime del .NET framework 2.0, conoscenze dei concetti di base dei linguaggi di markup

Software
Visual Studio 2005; .NET 3 Runtime; Visual Studio Extensions for Windows Presentation Foundation

Impegno

Tempo di realizzazione

Una delle novità introdotte dal Microsoft .NET Framework 3.0 è senz'altro Windows Presentation Foundation, il nuovo engine per la creazione di Interfacce Utente.
Oltre ad avanzate funzionalità di presentazione 3D, WPF, permette la creazione di interfacce utente molto più avanzate ed accattivanti rispetto agli strumenti già esistenti. WPF nasce, infatti, per sostituire l'attuale GDI che tutto sommato non ha grosse limitazioni ma ha il difetto di essere fortemente orientato al pixel e basato quindi su un modello ormai sorpassato. WPF mette a disposizione nuovi oggetti per disegnare oggetti vettoriali.
Per poter disegnare qualsiasi cosa con Windows Presentation Foundation possiamo utilizzare la libreria Shapes ovvero tutti gli oggetti contenuti nel namespace System.Windows.Shapes. Questo namespace contiene diverse classi ognuna relativa ad un diverso oggetto o forma grafica. Se, ad esempio, volessimo disegnare una semplice linea, utilizzeremo Line:

```
<Line X1="100" Y1="50" X2="400" Y2="100"
                              Stroke="Blue" />
```

Oppure per disegnare una forma circolare, ecco la classe Ellipse:

```
<Ellipse Width="100" Height="70" Stroke="Red"
                                  Fill="Red"  />
```

O ancora, se vogliamo ottenere un rettangolo, ecco la classe Rectangle:

```
<Rectangle Width="200" Height="110" Stroke=
      "Green"  Canvas.Left="50" Canvas.Top="80"/>
```

Inoltre, in Shapes, è contenuta la classe Path. L'oggetto Path è in assoluto il potente e completo dei precedenti e, in teoria, sarebbe possibile creare avanzate interfacce utilizzando solamente quest'ultimo oggetto con il solo svantaggio di non avere del codice molto prolisso rispetto a quello scritto con le più familiari classi Line, Ellipse o Rectangle.
Possiamo quindi disegnare un'ellisse anche utilizzando Path in questo modo:

```
<Path Fill="Red" Stroke="Red">
  <Path.Data>
    <EllipseGeometry Center="196,144" RadiusX=
                          "144" RadiusY="96"  />
  </Path.Data>
</Path>
```

Con queste semplici righe XAML, abbiamo ottenuto diverse forme base sul nostro form come possiamo vedere in Fig.1.

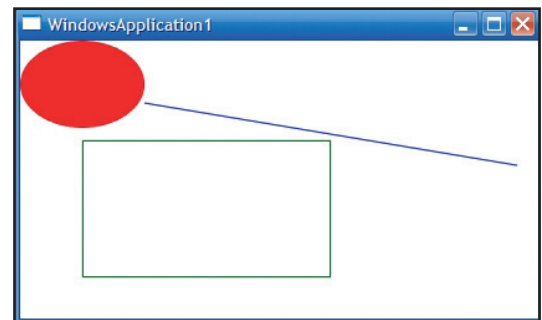



Figura 1: Alcuni oggetti Shapes in azione

Oltre alle solite proprietà per le dimensioni e posizioni della forma disegnata, queste classi hanno due proprietà di tipo Brush: Stroke e Fill. La prima indica il "tratto" da utilizzare per disegnare mentre la seconda indica il "riempimento" che può essere un semplice colore o qualcosa di più avanzato come ad esempio un'immagine bitmap.

DISEGNAMO UN OROLOGIO

Iniziamo, a questo punto, a disegnare qualcosa di più carino come ad esempio un orologio e, per rendere le cose più interessanti, realizzeremo un classico orologio con lancette
Inoltre, per utilizzare appieno le potenzialità vettoriali di WPF, non daremo delle dimensioni fisse alla figura ma queste saranno relative al ridimensionamen-

to del form da parte dell'utente in modo da apprezzare il fatto che, in qualsiasi risoluzione o qualsiasi tipo di display, il disegno non assume aspetti diversi.
Da Visual Studio selezioniamo File->New Project e, se abbiamo installato correttamente le estensioni per Windows Presentation Foundation, dovrebbe apparire nell'elenco dei tipi di progetto "NET Framework 3.0" e nei Templates "Windows Application (WPF)" come in Fig. 2.

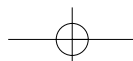
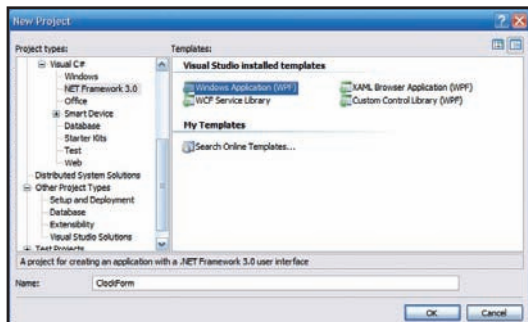



Figura 2: Creazione del nuovo progetto WPF in Visual Studio 2005

Apriamo il file Window1.xaml e creiamo l'elemento root nel nostro file XAML in questo modo:

```
<Window x:Class="ClockForm.Window1"
  xmlns="http://schemas.microsoft.com/winfx/
                                    2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/
                                              2006/xaml"
  xmlns:s="clr-namespace:System;assembly
                                              =mscorlib"
  Title="WPF Chronometer">

 <Window.Resources>
  <!-- Qui verranno definite le risorse necessarie al
                           funzionamento del nostro orologio.
Per adesso resterà vuoto -->
 </Window.Resources>

 <Viewbox>
  <Canvas Height="300" Width="300">
   <!-- Il content del Canvas sarà il nostro orologio
                                                     -->
  </Canvas>
 </Viewbox>

</Window>
```

Con queste righe abbiamo semplicemente definito un oggetto Window dove verrà disegnato il nostro orologio e, le uniche cose in più rispetto ad una finestra generata automaticamente da Visual Studio sono:

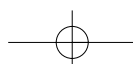
- La riga *xmlns:s="clr-namespace:System;assembly=mscorlib"*, definita per poter utilizzare delle classi che non sono dll XAML e, definendo il prefisso "s", questo renderà possibile l'accesso a tutte le classi e strutture presenti nel namespace System e dell'assembly mscorlib.dll;
- *<Window.Resources>…</Window.Resources>* dove successivamente verranno definite le risorse necessarie al funzionamente del nostro orologio, come ad esempio l'orario attuale;
- *<Viewbox>*, un controllo molto importante in WPF che, come vedremo, permette l'adattamento automatico del contenuto rispetto al contenitore. Nel nostro caso il nostro orologio (contenuto) verrà automaticamente ridimensionato al ridimensionamento della Window (contenitore);
- Infine il Panel *<Canvas>* che, tra tutti i tipi di Panel presenti in WPF, è il più indicato nel nostro caso in quanto permette il posizionamento assoluto dei controlli al suo interno.

Tutto il nostro orologio sarà, quindi, contenuto nell'oggetto Canvas.
Iniziamo a disegnare il bordo del nostro orologio per mezzo della classe Path che, come già descritto, è il più potente degli oggetti e possiamo, quindi, utilizzarlo per disegnare qualsiasi cosa .

Per prima cosa definiamo il bordo esterno:

```
   <!-- Bordo Esterno -->
   <Path Name="bordoEsterno" StrokeThickness=
                                              "10">
    <Path.Stroke>
     <LinearGradientBrush>
      <GradientStop Color="LightSlateGray"
                                        Offset="0" />
      <GradientStop Color="Silver" Offset="1" />
     </LinearGradientBrush>
    </Path.Stroke>
    <Path.Data>
     <EllipseGeometry Center="150, 150" RadiusX
                               ="120" RadiusY="120" />
    </Path.Data>
   </Path>
```

La proprietà più importante dell'oggetto Path è Data, di tipo Geometry, una classe astratta da cui derivano classi come LineGeometry, RectangleGeometry o EllipseGeometry. Attraverso queste classi è possibile disegnare qualsiasi cosa che verrà visualizzata da Path. Nel nostro caso abbiamo utilizzato un EllipseGeometry per disegnare, ovviamente, un cerchio impostando la proprietà Center (150, 150), ossia le coordinate del centro del nostro Canvas, e le proprietà RadiusX e RadiusY per impostare il raggio del nostro cerchio.
Il bordo del nostro orologio lo definiamo attraverso le proprietà Stroke e StrokeThickness che indicano, rispettivamente, il colore del bordo e la sua larghez-

za. Per ottenere un effetto sfumato abbiamo dovuto definire come Brush un LinearGradientBrush con colore di partenza LightSlateGray e colore finale Silver. A questo punto definiamo il bordo interno nella stessa maniera ma, per ottenere una sorta di effetto 3D, la direzione della sfumatura le definiremo nel senso inverso alla precedente:

```
<!-- Bordo interno -->
<Path Name="bordoInterno" StrokeThickness=
                                  "15" Fill="White">
  <Path.Stroke>
    <LinearGradientBrush>
      <GradientStop Color="Silver" Offset="0" />
      <GradientStop Color="LightSlateGray"
                                     Offset="1" />
    </LinearGradientBrush>
  </Path.Stroke>
  <Path.Data>
    <EllipseGeometry Center="150, 150" RadiusX
                          ="111" RadiusY="111">
    </EllipseGeometry>
  </Path.Data>
</Path>
```

A questo punto dovremmo avere già disegnato il bordo del quadrante come mostrato in Fig. 3.

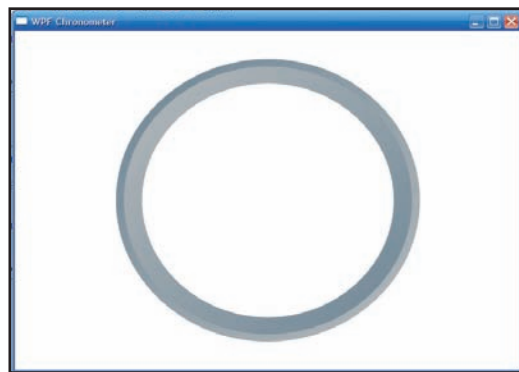



Figura 3: Il bordo del nostro orologio

Ma entriamo adesso nei dettagli del nostro orologio disegnando il quadrante, gli indicatori dell'orario e le lancette. Per disegnare il quadrante utilizzeremo un'ellisse nero appena più piccolo del bordo mentre, per gli indicatori, utilizzeremo la classe Line per creare delle tacche in corrispondenza delle ore 12, 3, 6 e 9.

```
<!-- Quadrante nero con tacchette blu -->
<Path Name="sfondoQuadrante" Fill="Black">
  <Path.Data>
    <EllipseGeometry Center="150,150" RadiusX
                           ="90" RadiusY="90" />
  </Path.Data>
</Path>
```

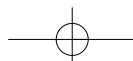
```
<Line X1="150" Y1="75" X2="150" Y2="70"
  Stroke="Blue" StrokeThickness="2" /><!-- 12 -->
<Line X1="225" Y1="150" X2="230" Y2="150"
   Stroke="Blue" StrokeThickness="2" /><!-- 3 -->
<Line X1="150" Y1="225" X2="150" Y2="230"
   Stroke="Blue" StrokeThickness="2" /><!-- 6 -->
<Line X1="75" Y1="150" X2="70" Y2="150"
   Stroke="Blue" StrokeThickness="2" /><!-- 9 -->
```

Abbiamo, ancora una volta, utilizzato un oggetto EllipseGeometry nella proprietà Data della classe Path per creare un cerchio ma, con la differenza di non aver utilizzato la proprietà Stroke per il bordo bensì Fill in modo da definire un riempimento uniforme dello sfondo del quadrante. Per le tacche relative all'orario abbiamo usato, questa volta, la classe Line che definisce una linea a partire dalle coordinate del punto iniziale (X1, Y1) fino al punto finale (X2, Y2) con il colore definito attraverso la property Stroke e la larghezza con StrokeThickness.
Passiamo adesso alle lancette che, come abbiamo fatto per le tacche, verranno disegnate utilizzando la classe Line in questo modo:

```
<!-- Lancetta delle ore -->
<Line Name="lancettaOre" X1="150" Y1="150"
                             X2="150" Y2="100"
     Stroke="Blue" StrokeThickness="6"
     StrokeEndLineCap="Triangle" StrokeStart
                           LineCap="Round">
</Line>

<!-- Lancetta dei minuti -->
<Line Name="lancettaMinuti" X1="150" Y1=
                    "150" X2="150" Y2="70"
     Stroke="Red" StrokeThickness="4"
     StrokeEndLineCap="Triangle" StrokeStart
                           LineCap="Round">
</Line>
<!-- Lancetta dei secondi -->
<Line Name="lancettaSecondi" X1="150" Y1=
                    "150" X2="150" Y2="70"
     Stroke="Yellow" StrokeThickness="2"
     StrokeEndLineCap="Triangle" StrokeStart
                           LineCap="Round">
</Line>
```

Abbiamo, in questo modo, disegnato le tre linee raffiguranti le lancette che partono dal centro del Canvas (150, 150) e arrivano all'indicatore delle ore 12 ad eccezione della lancetta delle ore che è, naturalmente, più corta. Per adesso ci accontentiamo, quindi, di posizionare le lancette verso le ore 12, successivamente vedremo come posizionarle sull'orario attuale. In questo utilizzo della classe Line, saltano all'occhio due nuove proprietà:

• StrokeEndLineCap – che indica come deve essere rap-

presentata la parte finale della linea
- StrokeStartLineCap – che indica, invece, come deve essere rappresentata la parte iniziale della linea

Le due property sono di tipo PenLineCap e possono assumere il valori:

- Flat – valore di default paragonabile ad una linea se inizio/fino particolare
- Round – disegna un semicerchio all'estremità della linea
- Square – disegna un rettangolo all'estremità con lo stesso spessore della linea
- Triangle – rappresenta le estremità con un triangolo equilatero con lunghezza pari allo spessore della linea

Nel nostro caso abbiamo usato per le tre linee un inizio (centro del Canvas) Round mentre, per la parte finale un triangolo in modo da ottenere un vero e proprio effetto lancetta.

IL POSIZIONAMENTO DELLE LANCETTE

A questo punto il nostro orologio è completo, se non per il fatto di essere statico e visualizzare sempre le ore 12. Vediamo adesso come, grazie alle nuove funzionalità di data binding, leggere l'orario attuale e rappresentare questo valore sul nostro orologio.
Il data binding in Windows Presentation Foundation è, finalmente, parte integrante dell'architettura dell'engine e non un contorno come lo è tuttora nel mondo Windows Forms.
Come illustrato nella **Fig.4**, il data binding è essenzialmente il collegamento tra "binding target" (destinazione del dato) e un "binding source" (sorgente).
La figura dimostra, inoltre, alcuni concetti fondamentali nel binding in Windows Presentation Foundation:

- Ogni operazione di binding coinvolge questi quattro parti: un oggetto binding target, una proprietà target, una sorgente binding, ed un percorso al valore nel binding source da usare.
- La proprietà target deve essere una "Dependency Property" una novità ed in WPF la maggior parte degli elementi che interagiscono con la UI ereditano indirettamente dalla classe *DependencyObject* la quale può essere vista come un contenitore di proprietà, in particolar modo di oggetti *DependencyProperty*.
- Anche se non specificato nella figura, l'oggetto sorgente di binding non deve per forza limitarsi ad oggetti dati del CLR ma supporta qualsiasi oggetto come, ad esempio, controlli, oggetti list, oggetto associato ad un datasource ADO.NET o web services, o un xmlnode che contiene dei dati.

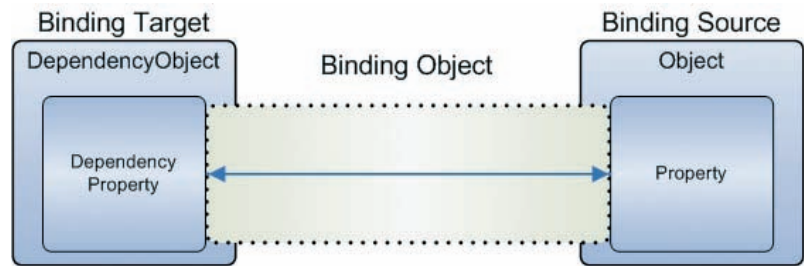



Figura 4: Il data binding in Windows Presentation Foundation

Un novità interessante è quella di poter agganciare tra loro i componenti visuali semplicemente usando l'attributo ElementName per indicare la fonte di binding:

```
<StackPanel>
  <TextBox Name="txt1" />
  <TextBox Name="txt2" Text="{Binding Element
          Name=txt1, Path=Text, Mode=OneWay}" />
</StackPanel>
```

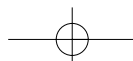
In questo esempio, la proprietà Text della TextBox txt2 viene collegata a quella di txt1 e, quindi, poiché la proprietà Text è una DependencyProperty, al variare del valore di txt1, verrà automaticamente aggiornato il valore di txt2. Da questo capiamo, quindi, che in alcune situazioni il data binding potrebbe sostituire benissimo l'utilizzo di eventi per la variazione di elementi nell'interfaccia utente.
Tornando al nostro orologio, potremmo utilizzare il data binding per leggere l'orario di sistema e rappresentarlo con le lancette e vediamo, quindi, come. Cominciamo innanzitutto a leggere la data di sistema. Nella sezione *Window.Resources* precedentemente definita, aggiungiamo la riga:

```
<!-- Legge la data e orario attuale -->
<FrameworkElement x:Key="dt" Tag="{x:Static s:
          DateTime.Now}" />
```

Utilizzando un FrameworkElement, abbiamo definito "dt" a cui è abbinato l'orario attuale attraverso DateTime.Now. Definendo, quindi, "dt" nella sezione Window.Resource, questa sarà disponibile in tutta la Window.
Per posizionare le lancette utilizzeremo gli oggetti TransformGroup che si occuperanno del calcolo necessario. Per esempio, l'angolo iniziale della lancetta delle ore deve essere impostata come somma tra 30 gradi per ogni ora e 0,5 gradi per ogni minuto. Ossia, per le 3:30 avremo 105 gradi come risultato di (3*30)+(30*0,5).
Nella sezione Window.Resource aggiungiamo il se-

NOTA

CHE COSA È UNA DEPENDENCY PROPERTY?

Una Dependency Property è una proprietà registrata con il nuovo sistema di Dependency Property di Windows Presentation Foundation. Infatti, vista la complessità dei controlli UI di WPF, che adesso possono essere anche formati da più controlli, questo sistema evita l'inutile instanza di proprietà non utilizzate. Maggiori approfondimenti all'indirizzo http://msdn2.microsoft.com/en-us/library/system.windows.dependencyproperty.aspx.

guente codice:

```
<!-- Moltiplica le ore per 30 gradi e i minuti per
0,5 gradi e somma.
Il risultato lo abbiamo in angleHour.
Value.OffsetX. -->
<TransformGroup x:Key="angleHour">
  <TranslateTransform
    X="{Binding Source={StaticResource dt},
    Path=Tag.Hour}"
    Y="{Binding Source={StaticResource dt},
    Path=Tag.Minute}" />
  <MatrixTransform Matrix="30 0 0.5 1 0 0" />
</TransformGroup>
```

In questo modo, attraverso l'uso di TranslateTransform, impostando X uguale all'ora e Y uguale ai minuti e moltiplicando tutto per una matrice con M11 uguale a 30 e M21 uguale a 0,5 avremo il risultato in OffsetX che verrà utilizzato come valore della posizione attuale della lancetta delle ore.
La definizione della lancetta delle ore sarà modificata quindi in questa maniera:

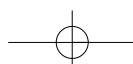
```
<!-- Lancetta delle ore -->
<Line Name="lancettaOre" X1="150" Y1="150"
X2="150" Y2="100"
    Stroke="Blue" StrokeThickness="6"
    StrokeEndLineCap="Triangle" StrokeStart
    LineCap="Round">
  <Line.RenderTransform>
    <RotateTransform x:Name="xformHour"
        Angle="{Binding Source=
  {StaticResource angleHour},Path=Value.OffsetX}"
        CenterX="150" CenterY="150"/>
  </Line.RenderTransform>
</Line>
```

Rispetto alla precedente definizione abbiamo, quindi, aggiunto una trasformazione alla Line attraverso la proprietà RenderTransform applicando una rotazione con angolo (Angle) uguale al valore di angleHour.Value.OffsetX definito nella sezione Resource dell'oggetto Window.
Un discorso analogo vale per le lancette dei minuti e secondi. Definiamo altri due oggetti TransformGroup adattati naturalmente alle angolature dei minuti e secondi.

```
<!-- Moltiplica i minuti per 6 gradi e i secondi per
0,1 gradi e somma.
Il risultato lo abbiamo in angleMinute.
Value.OffsetX. -->
<TransformGroup x:Key="angleMinute">
  <TranslateTransform
    X="{Binding Source={StaticResource dt},
    Path=Tag.Minute}"
    Y="{Binding Source={StaticResource dt},
    Path=Tag.Second}" />
  <MatrixTransform Matrix="6 0 0.1 1 0 0" />
</TransformGroup>

<!-- Moltiplica i secondi per 6 gradi. Il risultato lo
abbiamo in angleSecond.Value.M11. -->
<TransformGroup x:Key="angleSecond">
  <ScaleTransform
    ScaleX="{Binding Source={StaticResource
    dt}, Path=Tag.Second}" />
  <ScaleTransform ScaleX="6" />
</TransformGroup>
```

Dopo aver definito, sempre nella sezione Resource, i **due nuovi oggetto** TransformGroup, colleghiamo l'angolatura delle lancette dei minuti e secondi con i valori restituiti dai TransformGroup.
Modifichiamo, quindi, la definizione delle due lancette in questo modo:

```
<!-- Lancetta dei minuti -->
<Line Name="lancettaMinuti" X1="150" Y1=
"150" X2="150" Y2="70"
    Stroke="Red" StrokeThickness="4"
    StrokeEndLineCap="Triangle" StrokeStart
    LineCap="Round">
  <Line.RenderTransform>
    <RotateTransform x:Name="xformMinute"
        Angle="{Binding Source={Static
    Resource angleMinute}, Path=Value.OffsetX}"
        CenterX="150" CenterY="150"/>
  </Line.RenderTransform>
</Line>

<!-- Lancetta dei secondi -->
<Line Name="lancettaSecondi" X1="150" Y1=
"150" X2="150" Y2="70"
    Stroke="Yellow" StrokeThickness="2"
    StrokeEndLineCap="Triangle" StrokeStart
    LineCap="Round">
  <Line.RenderTransform>
    <RotateTransform x:Name="xformSecond"
        Angle="{Binding Source=
{StaticResource angleSecond}, Path=Value.M11}"
        CenterX="150" CenterY="150"/>
  </Line.RenderTransform>
</Line>
```

Come abbiamo già fatto per la lancetta delle ore, aggiungendo una trasformazione alle due lancette, l'angolatura corrispondente verrà applicata in base all'orario attuale.
Mandando in esecuzione la nostra applicazione, vedremo come le lancette verranno posizionate nelle rispettive coordinate anche se il lavoro non è ancora completo. Infatti, le nostre lancette saranno comunque statiche e visualizzeranno solamente l'orario letto in fase di caricamento.

Vediamo, quindi, come usare le potenti funzionalità di animazione di Windows Presentation Foundation.

ANIMIAMO IL TUTTO

Le animazioni di WPF non sono poi così diverse da quelle ottenibili nelle applicazioni WinForm, entrambe si ottengono modificando dinamicamente il valore di una o più proprietà di un elemento di UI, quello che distingue WPF è il mettere a disposizione una serie di classi che rendono la modifica più semplice rispetto all'equivalente Windows Forms 2.0.
Le classi di animazione sono divise in due gruppi e utilizzano un naming ben preciso: <*Type*>*Animation* e <*Type*>*AnimationUsingKeyFrames* dove <Type> rappresenta il tipo della proprietà da animare.
Se, ad esempio, vogliamo applicare un'animazione in modo che venga modificata la proprietà width di un button (che è di tipo Double) utilizzeremo DoubleAnimation.
Le animazioni possono essere descritte con XAML, in questo caso però l'animazione deve essere racchiusa in una *Timeline* che, come per altro l'animazione stessa, rappresenta un intervallo di tempo, all'interno del quale possiamo animare contemporaneamente più proprietà e affinché questo sia possibile dobbiamo racchiudere le animazioni in un contenitore di Timelines/Animazioni rappresentato dall'oggetto *StoryBoard*.
Una volta definita l'animazione va specificato come questa deve partire, ecco perché le animazioni sono normalmente contenute all'interno di un EventTrigger anche se posso essere contenute anche in uno style.
Supponiamo di avere un pulsante che, al click, deve ingrandirsi, ma questo deve avvenire non di scatto da una dimensione all'altra, ma gradualmente in modo da ottenere un effetto di animazione.

```
<StackPanel Orientation="Vertical" Name="sp1">
  <StackPanel.Triggers>
    <EventTrigger RoutedEvent="Button.Click"
                            SourceName="btnAnimate">
      <EventTrigger.Actions>
        <BeginStoryboard>
          <Storyboard>
            <DoubleAnimation From="100" To="200"
                                        Duration="0:0:2"
                          AutoReverse="True"
                          RepeatBehavior="Forever"
                       Storyboard.TargetName="btnAnimate"
      Storyboard.TargetProperty="Width" />
          </Storyboard>
        </BeginStoryboard>
      </EventTrigger.Actions>
    </EventTrigger>
  </StackPanel.Triggers>
  <Button Name="btnAnimate" Width="100" Height
                  ="70" Content="ClickMe!"/>
</StackPanel>
```

In questo modo abbiamo definito una animazione sulla proprietà Width del controllo btnAnimate da un valore di partenza uguale a 100 fino ad arrivare a 200 in un periodo di tempo pari a 2 secondi. Tornando al nostro orologio, dovremmo semplicemente definire tre animazioni, una per ogni lancetta e, naturalmente, andare ad intervenire sulla proprietà Angle della trasformazione definita precedentemente.

```
<!-- Animazioni delle lancette -->
<Window.Triggers>
  <EventTrigger RoutedEvent="Canvas.Loaded">
    <BeginStoryboard>
      <Storyboard>
 <DoubleAnimation Storyboard.TargetName=
                                    "xformHour"
          Storyboard.TargetProperty="Angle"
          From="0" To="360" Duration="12:0:0"
          IsAdditive="True"
          RepeatBehavior="Forever" />

 <DoubleAnimation Storyboard.TargetName=
                                  "xformMinute"
          Storyboard.TargetProperty="Angle"
          From="0" To="360" Duration="1:0:0"
          IsAdditive="True"
          RepeatBehavior="Forever" />
 <DoubleAnimation Storyboard.TargetName=
                                  "xformSecond"
          Storyboard.TargetProperty="Angle"
          From="0" To="360" Duration="0:1:0"
          IsAdditive="True"
          RepeatBehavior="Forever" />

 </Storyboard>
 </BeginStoryboard>
 </EventTrigger>
</Window.Triggers>
```

Ogni lancetta effettuerà, quindi, una rotazione di 360 gradi ognuna della durata corrispondente ossia la lancetta dei secondi in 60 secondi, dei minuti in 60 minuti mentre la lancetta delle ore in 12 ore.
Volendo, possiamo anche eliminare il bordo e sfondo standard della finestra attraverso le proprietà WindowStyle, AllowsTransparency e Background dell'oggetto Window:

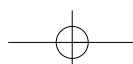
```
WindowStyle="None" AllowsTransparency="True"
Background="Transparent"
```

Vito Arconzo

UN PO' DI MATEMATICA

La classe MatrixTransform crea una trasformazione matematica matrice usata per manipolare oggetti o coordinate in un piano 2D. Maggiori approfondimenti all'indirizzo http://msdn2.microsoft.com/en-us/library/system.windows.media.matrixtransform.aspx.

VITO ARCONZO

sviluppa applicazioni sia Windows Forms che Web basate sul Microsoft .NET Framework già dalle prime versioni beta. In questo momento ciò che lo appassiona maggiormente è il nuovo engine introdotto con la versione 3.0 del .NET Framework, Windows Presentation Foundation e, tra le altre cose, collabora con lo User Group DotNetSide (www.dotnetside.org)

I trucchi del mestiere

Tips & Tricks

Questa rubrica raccoglie trucchi e piccoli pezzi di codice, frutto dell'esperienza di chi programma, che solitamente non trovano posto nei manuali. Alcuni di essi sono proposti dalla redazione, altri provengono da una ricerca su Internet, altri ancora ci giungono dai lettori. Chi volesse contribuire, potrà inviare i suoi Tips&Tricks preferiti. Una volta selezionati, saranno pubblicati nella rubrica. Il codice completo dei tips è presente nel CD allegato nella directory *tips*\\ o sul Web all'indirizzo: cdrom.ioprogrammo.it.

POWERSHELL

RILEVARE I CODEC AUDIO E VIDEO INSTALLATI

Questo piccolo script consente di rilevare le informazioni che riguardano i codec audio e video installati sul sistema. Come qualcuno potrà facilmente verificare analizzando la classe *Win32_CodecFile*, tali dati non sono gli unici che risulta possibile ottenere. Quelli mostrati sono quelli che potrebbero tornarci maggiormente utili. Per una lista esaustiva di tutte le informazioni della classe sopraccitata, è possibile riferirsi al link ***http://windowssdk.msdn.microsoft.com/en-us/library/ms753639.aspx***

```
$strComputer = "."
$Conteggio = 0

Function ListaCodec
  {
  param([string]$GroupType)

  # Imposta la stringa che funge da filtro
     $Filtro="Group = '$GroupType'"

  # Definisci la query che consente di estrarre le informazioni
                                      riguardanti i codec installati
     $colItems = get-wmiobject -class "Win32_CodecFile" -
                                      namespace "root\CIMV2" `
                     -filter $Filtro -computername $strComputer

write-host "ELENCO CODEC $GroupType"

ForEach ($objItem in $colItems)
  {

     $Conteggio = $Conteggio + 1

     write-host "-----------------------
                          --------------------------------
                                                    --------"
     write-host "Descrizione: " $objItem.Description
     write-host "Data installazione: " $objItem.InstallDate
     write-host "Manufacturer: " $objItem.Manufacturer
     write-host "Nome: " $objItem.Name
     write-host "Percorso: " $objItem.Path
     write-host "Versione: " $objItem.Version

  }
  write-host "----------------------------
                               -----------------------------
                                                       -----"
  write-host "TOTALE CODEC RILEVATI: $Conteggio"
  write-host "----------------------------
                               -----------------------------
                                                       -----"
}

# Elenca le informazioni sui codec video
  ListaCodec "Video"
# Elenca le informazioni sui codec audio
  ListaCodec "Audio"
```

ATTENDERE LA DISTRUZIONE DI UN PROCESSO

Con uno script PowerShell è piuttosto banale controllare quando un processo viene distrutto. Tutto quello che occorre fare è definire il nome del processo da tenere sott'occhio ed utilizzare la *WaitForExit()*. In questo esempio, abbiamo preso in considerazione il processo Task Manager. Per poterlo controllare, viene dapprima letta la variabile di ambiente *SystemRoot* per identificare il giusto percorso dell'applicativo e, successivamente, dopo averlo lanciato attraverso la cmdlet *Invoke-Item*, si raggiunge l'obiettivo prefissato.

```
# Leggi la variabile SystemRoot e memorizzala
  $PathToWindows = $Env:SystemRoot

# Crea il path completo della sottocartella System32...
  $PathToApplication = $PathToWindows + "\System32"

# Lancia il processo Task Manager
  Invoke-Item "$PathToApplication\taskmgr.exe"

# Imposta la variabile che indica il processo Task Manager da
```

```
                                                       controllare
  $ProcessoDaControllare = get-process taskmgr

# Attendi che il processo venga chiuso per poter inviare un
                                                 messaggio a video
  clear
  write-host "Chiudi la finestra Task Manager per visualizzare
                                                     un messaggio."
  $ProcessoDaControllare.WaitForExit()
# Visualizza il messaggio
  clear

  write-host "Il processo Task Manager è stato distrutto. Lo
                                          script può continuare!"
```

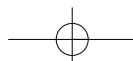
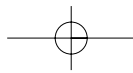
TENERE SOTTO CONTROLLO L'AGGIORNAMENTO

Questo script consente di tenere sotto controllo l'aggiornamento dei file contenuti all'interno di una qualunque cartella. Esso può tornare utile per controllare, ad esempio, se ci sono file di log che non vengono aggiornati ad intervalli di tempo predefiniti (nell'esempio abbiamo considerato ogni 30 minuti). In particolare, l'istruzione *(get-date).AddMinutes(-$INTERVAL)* consente calcolare la data e l'ora corrispondente a 30 minuti prima del momento in cui lo script viene lanciato. Si osservi, infine, che esistono altrettanti metodi per aggiungere/togliere ore, giorni, anni, ecc. definiti analogamente come *AddHours*, *AddDays*, *AddYears*, ecc.

```
# Preleva tutti i files nella directory C:\LOG aventi l'estensione
                                                        desiderata

 $FSO = new-object –com Scripting.FileSystemObject
 $folderObject = $FSO.GetFolder("C:\LOG")
 $filesObject = $folderObject.Files
 $Estensione = "txt"

# Definisci un intervallo di controllo (per esempio 10 min.)
 $INTERVAL = 10

# Dalla collection files controlla se ci sono file aventi
                                          l'estensione desiderata

# Stampa l'elenco dei file che non sono stati aggiornati
                                       nell'intervallo specificato
  write-host "ELENCO FILE NON AGGIORNATI"
# Un risultato analogo lo si poteva ottenere anche mediante il
                               seguente comando, opportunamente
# modificato in base alle esigenze:
# dir "$folderObject\*.txt"| where-object
                                            { $_.LastWriteTime -ge
                              (get-date).AddMinutes(-$INTERVAL) }

 ForEach($FileTrovato In $FilesObject)
     {
     # Se ci sono file con l'estensione scelta

         If($FSO.GetExtensionName($FileTrovato.Name) -eq
                                                     $Estensione)
           {
            # Controlla la data di ultima modifica e l'ora che non
sia
                                            superiore a INTERVAL
            # Il parametro "n" della DateDiff specifica che la
                                    differenza è espressa in MINUTI

              If($FileTrovato.DateLastModified -le (get-
                                 date).AddMinutes(-$INTERVAL))
                {
                # Scatta allarme
                  write-host $FileTrovato.Name
                }
           }
      }
```

UTILIZZARE L'OGGETTO INTERNET EXPLORER

La PowerShell consente di referenziare ed utilizzare oggetti COM. Per far questo ci si serve della New-Object che, come vedremo, risulta relativamente semplice da utilizzare. In quest'esempio, apriremo una finestra Internet Explorer ed imposteremo alcune sue proprietà. In particolare, attraverso l'istruzione *$Int Explorer|get-member* possiamo visualizzare a video l'elenco di tutte le proprietà e dei metodi supportati.

```
# Crea il nuovo oggetto
  $IntExplorer = New-Object -ComObject
                                  InternetExplorer.Application

# Elenchiamo tutti i metodi e le proprietà dell'oggetto
  write-host "Elenco delle proprietà dell'oggetto
                                        INTERNET.APPLICATION"
  write-host
       "*********************************************"
  $IntExplorer|get-member
  write-host
"**************************************************"

# Utilizziamo alcuni dei metodi e delle proprietà
# Impostiamo la pagina web

  $IntExplorer.Navigate("http://www.ioProgrammo.it")

# Attiva la finestra a Full Screen
  $IntExplorer.Visible=1
  $IntExplorer.FullScreen=0

# Posiziona la finestra di Internet Explorer
  $IntExplorer.Top=0
  $IntExplorer.Left=0
# Fa sparire la barra dei menu
  $IntExplorer.MenuBar=0
```

UN MIGRATORE DI DATI UNIVERSALE

VI E' MAI CAPITATO DI IMPLEMENTARE IL VOSTRO PROGRAMMA SU UN SISTEMA DI PROVA E DOVERLO POI IMPLEMENTARE IN UN SISTEMA IN PRODUZIONE? LA MIGRAZIONE DELLA STRUTTURA DATI POTREBBE NON ESSERE COSÌ SEMPLICE. ECCO UN PICCOLO AIUTO...

REQUISITI

Conoscenze richieste
Mincrosoft.Net, T-SQL

Software
Visual Studio 2005, SQL Server 2005

Impegno

Tempo di realizzazione

Quante volte ci è capitato di voler trasferire il contenuto di tabelle da un database ad un altro e non abbiamo potuto (o voluto) utilizzare strumenti automatici come le funzioni di importazione ed esportazione dati di SQL Server? In quante occasioni ci troviamo a dover importare (anche più volte nella stessa giornata) dei dati inseriti nelle nostre tabelle da un database di sviluppo ad uno di produzione o viceversa?
Il programma che svilupperemo nel corso di questo articolo può risultare molto utile in queste situazioni.
La funzione principale del software che andremo a realizzare è la seguente: leggere il contenuto di una tabella di SQL Server e generare uno script T-SQL per l'insert dei dati in una tabella/database di destinazione.
Prendiamo come esempio la tabella in figura:
Nel corso dell'articolo vedremo come sia possibile realizzare un programma in grado di generare uno script di questo tipo:

Table - dbo.Categories | Table - dbo.Regions | Summary

	cat_id	cat_short_name	cat_name
▶	1	telefonia	Telefonia
	2	audio	Audio
	3	video	Video
*	NULL	NULL	NULL

Fig. 1: La tabella da esportare

```
SET IDENTITY_INSERT [categories] ON
INSERT INTO [categories]  ([cat_id], [cat_short_na
  me], [cat_name])  VALUES  (1, 'telefonia', 'Telefonia')
INSERT INTO [categories]  ([cat_id], [cat_short_na
        me], [cat_name])  VALUES  (2, 'audio', 'Audio')
INSERT INTO [categories]  ([cat_id], [cat_short_na
        me], [cat_name])  VALUES  (3, 'video', 'Video')
SET IDENTITY_INSERT [categories] OFF
```

Utilizzeremo delle tabelle e delle viste di sistema di SQL Server ed un po' di fantasia…
Vediamo, innanzitutto, quali sono gli strumenti a nostra disposizione.

SYSOBJECTS E VISTE TABLES

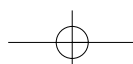
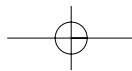
SQL Server memorizza in alcune tabelle di sistema tutte le informazioni sulla struttura del database, sulle tabelle che abbiamo creato (con relative colonne), sulle viste, stored procedure e funzioni -tanto per citare qualche esempio.
Per quanto riguarda le tabelle abbiamo due possibilità; la prima:

```
SELECT [Name] AS TABLE_NAME
FROM sysobjects
WHERE
  [type] = 'u'
ORDER BY [Name]
```

Legge i dati dalla vista di sistema "sysobjects". Filtrando i dati per type = 'u' ci vengono restituite soltanto le tabelle utente.
Useremo i seguenti valori per identificare il tipo di oggetto che ci interessa:

- ***U***: tabelle utente
- ***P***: Stored procedure
- ***TR***: Trigger
- ***V***: viste
- ***PK***: chiavi primarie
- ***C***: vincoli
- ***D***: defaults
- ***F***: chiavi esterne

Per quanto riguarda le colonne possiamo utilizzare due oggetti syscolumns per le colonne, systypes per ottenere il tipo di dato della colonna.

```
SELECT
  syscolumns.[name] AS COLUMN_NAME,
  systypes.[name] AS DATA_TYPE
FROM syscolumns
INNER JOIN sysobjects
```

```
ON syscolumns.id = sysobjects.id
INNER JOIN systypes
 ON syscolumns.xtype = systypes.xtype
WHERE sysobjects.[name] = 'Products'
```

Un altro metodo è utilizzare le viste INFORMATION_SCHEMA.

Ad esempio, la ***INFORMATION_SCHEMA.TABLES*** restituisce la lista delle tabelle mentre la ***INFORMATION_SCHEMA.ROUTINES*** restituisce la lista di stored procedures e funzioni.

Per visualizzare la lista delle tabelle, scriveremo:

```
SELECT TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE'
 AND TABLE_NAME <> 'sysdiagrams'
ORDER BY TABLE_NAME
```

Mentre per le colonne della singola tabella:

```
SELECT
 COLUMN_NAME,
 DATA_TYPE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE
 table_name = 'Products'
```

In questo ultimo esempio scorriamo le tabelle del nostro database e per ogni tabella trovata stampiamo a video nome e tipo di tutte le colonne.

```
DECLARE @TABLE_NAME NVARCHAR(128),
        @COLUMN_NAME
                NVARCHAR(128),
        @DATA_TYPE NVARCHAR(128)

-- Scorriamo le tabelle
DECLARE curTable CURSOR FOR
        SELECT TABLE_NAME
        FROM INFORMATION_SCHEMA.TABLES
        WHERE TABLE_TYPE = 'BASE TABLE'
        AND TABLE_NAME <> 'sysdiagrams'
        ORDER BY TABLE_NAME
OPEN curTable

FETCH NEXT FROM curTable INTO @TABLE_NAME

WHILE @@FETCH_STATUS = 0
BEGIN

        PRINT @TABLE_NAME
        PRINT '************************'

        -- Scorriamo le colonne della tabella
        DECLARE curColumns CURSOR FOR
                SELECT COLUMN_NAME,
                        DATA_TYPE
                FROM
                INFORMATION_SCHEMA.COLUMNS
                WHERE
                 table_name = @TABLE_NAME

                OPEN curColumns

                FETCH NEXT FROM curColumns
                INTO @COLUMN_NAME, @DATA_TYPE

                WHILE @@FETCH_STATUS = 0
                BEGIN

                        PRINT @COLUMN_NAME
                        + ': ' + @DATA_TYPE

                        FETCH NEXT FROM
        curColumns INTO @COLUMN_NAME, @DATA_TYPE
                END

                CLOSE curColumns
                DEALLOCATE curColumns

        PRINT ''
        PRINT ''

        FETCH NEXT FROM curTable INTO
                @TABLE_NAME
END
CLOSE curTable
DEALLOCATE curTable
```

Adesso abbiamo le conoscenze di base per scrivere il nostro software, mettiamoci all'opera!

IL PROGETTO

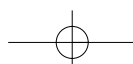
Creiamo un nuovo progetto di tipo "***Console Application***" con Visual Studio e chiamiamolo "TablesToSQLDemoVB".
Creiamo il file di configurazione "App.config" e definiamo due parametri: la stringa di connessione al database (in cui dovrete inserire il nome del vostro server), ed un parametro "*queryType*" che farà utilizzare al programma gli oggetti schema oppure gli oggetti sys in base al suo valore.
Trovate il file nel CD allegato.
Apriamo il file ***Module1.vb*** che rappresenta l'entry point del programma.
Dichiariamo due variabili globali:

```
Dim queryType As String
Dim Conn As SqlConnection
```

la prima è legata al parametro dell'App.config, la

NOTA

IL CODICE ALLEGATO

Gli esempi nell'articolo sono in Visual Basic; per chi fosse interessato, nel CD allegato è presente tutto il codice sorgente anche in C#.

seconda contiene la connessione al database. Analizziamo il ***Main***:

```
Sub Main(ByVal args() As String)
    queryType =
ConfigurationManager.AppSettings("databaseVersion")
    Dim connectionString As String = ""

 ' vedo se ho passato dei parametri al programma
    If (args.Length = 2) Then
        Dim server As String = args(0) 'primo
                                        parametro: il server
        Dim catalog As String = args(1) 'secondo
                                        parametro: il catalogo
        connectionString = _
            "Data Source=" & server & _
            ";Initial Catalog=" & catalog & _
            ";Integrated Security=True"" provider
                            Name=""System.Data.SqlClient"
    Else
        connectionString = _
ConfigurationManager.ConnectionStrings("Connection
                            String").ConnectionString
    End If

    ' istanzio la connessione

    Conn = New SqlConnection(connectionString)
    ' Aspetto un comando
    Dim fullCommand As String = ""
    Do
        Console.Write("$$$ ")
        fullCommand =
                    Console.ReadLine().ToLower().Trim()
        ExecuteCommand(fullCommand)
    Loop While (fullCommand <> "quit")
End Sub
```

La prima parte del metodo legge i parametri dall'App.config ed istanzia la connessione.
Se abbiamo passato due parametri alla riga di comando, i nomi del database e del server vengono letti da questi, altrimenti dal file di configurazione. Dopo aver istanziato la connessioni, il programma si comporta come un ***listener***, restando in attesa di un comando.
La funzione ***ExecuteCommand*** legge il nostro comando e lo interpreta; il funzionamento è semplice e non lo analizzeremo nel corso dell'articolo. Il codice è comunque presente nel CD allegato insieme a resto del progetto.

VISUALIZZIAMO LE TABELLE DEL DATABASE

Analizziamo la funzione che ci permette di ottenere la lista delle tabelle:

System Views
CRI\SQL2005\Databases\Products\Views\System Views

Name	Schema	Created
CHECK_CONSTRAINTS	INFORMATION_SCHEMA	14/10/2005
COLUMN_DOMAIN_USAGE	INFORMATION_SCHEMA	14/10/2005
COLUMN_PRIVILEGES	INFORMATION_SCHEMA	14/10/2005
COLUMNS	INFORMATION_SCHEMA	14/10/2005
CONSTRAINT_COLUMN_USAGE	INFORMATION_SCHEMA	14/10/2005
CONSTRAINT_TABLE_USAGE	INFORMATION_SCHEMA	14/10/2005
DOMAIN_CONSTRAINTS	INFORMATION_SCHEMA	14/10/2005
DOMAINS	INFORMATION_SCHEMA	14/10/2005
KEY_COLUMN_USAGE	INFORMATION_SCHEMA	14/10/2005
PARAMETERS	INFORMATION_SCHEMA	14/10/2005
REFERENTIAL_CONSTRAINTS	INFORMATION_SCHEMA	14/10/2005
ROUTINE_COLUMNS	INFORMATION_SCHEMA	14/10/2005
ROUTINES	INFORMATION_SCHEMA	14/10/2005
SCHEMATA	INFORMATION_SCHEMA	14/10/2005
TABLE_CONSTRAINTS	INFORMATION_SCHEMA	14/10/2005
TABLE_PRIVILEGES	INFORMATION_SCHEMA	14/10/2005
TABLES	INFORMATION_SCHEMA	14/10/2005
VIEW_COLUMN_USAGE	INFORMATION_SCHEMA	14/10/2005
VIEW_TABLE_USAGE	INFORMATION_SCHEMA	14/10/2005
VIEWS	INFORMATION_SCHEMA	14/10/2005

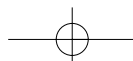
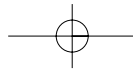



Fig. 2: Le tabelle di sistema che memorizzano la struttura del DB

```
Private Sub ListTables()
    Dim tablesList As List(Of String) = getTablesList()
    For Each table As String In tablesList
        Console.WriteLine(table)
    Next
    Console.WriteLine("")
End Sub

Private Function getTablesList() As List(Of String)

    Dim tablesList As List(Of String) = New List
                                        (Of String)
    Try

        Conn.Open()
        Dim Comm As SqlCommand =
                            New SqlCommand()
        Comm.Connection = Conn
        Comm.CommandText = GetTablesSQL()

        Dim Reader As SqlDataReader =
                            Comm.ExecuteReader()
        Try
            While Reader.Read()
                TablesList.Add(Reader("TABLE_NAME").
                                        ToString())
            End While
        Finally
            Reader.Close()
        End Try
    Finally
        Conn.Close()
    End Try
    Return TablesList

End Function
```

La funzione ***ListTables*** istanzia una lista e la po-

pola tramite la *getTablesList*, poi stampa a video i nomi delle tabelle.
La funzione ***TablesList*** utilizza la ***GetTablesSQL*** per ottenere la query da eseguire al fine di recuperare la lista delle tabelle ed un *DataReder* per popolare la lista con i nomi delle tabelle.
Ecco il codice:

```
Private Function GetTablesSQL() As String
    If queryType = "schema" Then
        Return "SELECT TABLE_NAME" & _
            " FROM INFORMATION_SCHEMA.TABLES" &
                                                   _
            " WHERE TABLE_TYPE = 'BASE TABLE'" & _
            "   AND TABLE_NAME <> 'sysdiagrams'" & _
            " ORDER BY TABLE_NAME"
    Else
        Return "SELECT [Name] AS TABLE_NAME" & _
            " FROM sysobjects" & _
            " WHERE" & _
            "   [type] = 'u'" & _
            " ORDER BY [Name]"
    End If
End Function
```

Per testare la funzione, al prompt digitiamo "list" e premiamo invio.

CREIAMO GLI SCRIPT

Se al prompt digitiamo "sql:nometabella", viene lanciata la funzione GenerateSQL, che genera lo script SQL contenente le istruzioni di insert per scrivere i dati nella tabella.
Vediamo come funziona:

```
Private Sub GenerateSQL(ByVal arrCommand() As
                                        String)
    Dim tableColumns As List(Of TableColumn)
    Dim sqlScripts As List(Of String)

    '1. Lettura parametri
    Dim paramsNumber As Integer =
                          arrCommand.Length - 1
    If paramsNumber < 1 Then
        Console.WriteLine("list: nome tabella mancante")
        Return
    End If
    If paramsNumber > 1 Then
        Console.WriteLine("list: troppi parametri")
        Return
    End If

    '2. nome tabella e file su cui salvare lo script
    Dim tableName As String = arrCommand(1)
    Dim fileName As String = tableName & ".txt"
```

```
    Dim f As StreamWriter = Nothing
    f = File.CreateText(fileName)

    '3. generazione script
    Try
        Conn.Open()
        If (TableExists(Conn, tableName)) Then '3.1.
                          verifico l'esistenza della tabella
            '3.2. lettura colonne
            tableColumns = GetColumnList(tableName,
                                                  Conn)

            '3.3. lettura dati e preparazione script
            sqlScripts = GetSQLStatements(tableName,
                                   Conn, tableColumns)

            '3.4. scttura dati
            SaveSQLScript(tableName, fileName, f,
                                             sqlScripts)

        Else
            Console.WriteLine("sql: tabella inesistente")
            Return
        End If
    Finally
        'rilascio risorse
        Conn.Close()
        f.Close()
    End Try
    '4. apertura file
    System.Diagnostics.Process.Start(fileName)
End Sub
```

Fig. 2: Flow Chart del metodo

NOTA

SYSTEM. COLLECTIONS. GENERIC

È un nuovo namespace introdotto con .Net 2.0. Contiene i "Generics" delle collection tipizzate molto potenti e performanti. Nel programma abbiamo utilizzato le List, delle liste tipizzate. La forza dei generics risiede nel fatto che, conoscendo a priori il tipo di dato che contengono, non sono necessarie le operazioni di boxing ed unboxing che erano invece indispensabili, ad esempio, con gli ArrayList, i cui elementi sono degli object.

Come prima cosa leggiamo il parametro rappresentante il nome della tabella da esportare; creiamo quindi il file su cui salvare lo script.
Lo step di generazione script:

- verifica l'esistenza della tabella da esportare
- legge quali sono le colonne della tabella
- legge i dati dalla tabella e prepara lo script
- salva lo script su file

Analizziamo i passi significativi uno alla volta. Tralasciamo la funzione di verifica esistenza tabella, molto semplice - comunque presente nel CD allegato.
Per la lettura ed il salvataggio in memoria della lista delle colonne utilizziamo una classe interna.

```
Private Class TableColumn

    Private _COLUMN_NAME As String
    Public ReadOnly Property COLUMN_NAME() As
                                        String
        Get
            Return _COLUMN_NAME
        End Get
    End Property

    Private _DATA_TYPE As String
    Public ReadOnly Property DATA_TYPE() As String
        Get
            Return _DATA_TYPE
        End Get
    End Property

    Public Sub New(ByVal COLUMN_NAME As String,
                            ByVal DATA_TYPE As String)
        _COLUMN_NAME = COLUMN_NAME
        _DATA_TYPE = DATA_TYPE
    End Sub
End Class
```

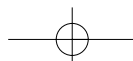
Ci consentirà di memorizzare in una lista generica le coppie di valori "nome colonna" / "tipo di dato". Vediamo la funzione ***GetColumnList*** che legge i nomi delle colonne e li salva in una lista.

```
Private Function GetColumnList(ByVal tableName As
        String, ByVal Conn As SqlConnection) As List(Of
                                        TableColumn)
    Dim tableColumns As List(Of TableColumn) = New
                                    List(Of TableColumn)

    Dim Comm As SqlCommand = New SqlCommand()
    Comm.Connection = Conn
    Comm.CommandText = GetColumnsSQL
                                        (tableName)

    Dim Reader As SqlDataReader =
                            Comm.ExecuteReader()
    Try

        Do While (Reader.Read())
            tableColumns.Add(New TableColumn
                (Reader("COLUMN_NAME").ToString(), _
                Reader("DATA_TYPE").ToString()))
        Loop
    Finally
        Reader.Close()
    End Try

    Return tableColumns
End Function
```

Viene istanziata una lista generica *List<TableColumn>* di oggetti di tipo *TableColumn*, gli elementi vengono quindi aggiunti alla lista tramite un *DataReader* che legge il risultato della query.
Per utilizzare le liste generiche è necessario importare un namespace: ***System.Collection.Generics***

```
Private Function GetColumnsSQL(ByVal tableName As
                                String) As String
    If queryType = "schema" Then
        Return "SELECT" & _
            "  COLUMN_NAME," & _
            "  DATA_TYPE" & _
            " FROM INFORMATION_SCHEMA.COLUMNS"
                                            & _
            " WHERE" & _
            "  table_name = '" + tableName + "'"
    Else
        Return "SELECT" & _
            "   syscolumns.[name] AS COLUMN_NAME,"
                                            & _
            "   systypes.[name] AS DATA_TYPE" & _
            " FROM syscolumns" & _
            " INNER JOIN sysobjects" & _
            "   ON syscolumns.id = sysobjects.id" & _

            " INNER JOIN systypes" & _
            "  ON syscolumns.xtype = systypes.xtype" &
    " WHERE sysobjects.[name] = '" + tableName + "'"
    End If
End Function
```

Vediamo quindi la funzione che prepara lo script:

```
Private Function GetSQLStatements(ByVal tableName
    As String, ByVal Conn As SqlConnection, ByVal table
    Columns As List(Of TableColumn)) As List(Of String)

    Dim sqlScripts As List(Of String) = New List
                                        (Of String)
```

```
Dim Comm As SqlCommand = New SqlCommand()
Comm.Connection = Conn
Comm.CommandText = "SELECT * FROM "
                                        & tableName

Dim Reader As SqlDataReader =
                              Comm.ExecuteReader()
Try
   Do While (Reader.Read())

      Dim sbSQL As StringBuilder =
                                New StringBuilder()
      sbSQL.Append("INSERT INTO
                           [" & tableName & "] ")

      'colonne

      Dim sbColumns As StringBuilder = New
                                   StringBuilder()
      For Each tc As TableColumn In tableColumns

         sbColumns.Append("[" & tc.COLUMN_NAME
                                          & "], ")
      Next
      sbColumns.Remove(sbColumns.Length - 2, 2)
      sbSQL.Append(" (")

      sbSQL.Append(sbColumns)
      sbSQL.Append(") ")

      sbSQL.Append(" VALUES ")

       'valori
      Dim sbColumnVaules As StringBuilder = New
                                   StringBuilder()
      For Each tc As TableColumn In tableColumns
         sbColumnVaules.Append(FormatInput
                      (Reader(tc.COLUMN_NAME), _
         tc.DATA_TYPE) + ", ")
      Next
sbColumnVaules.Remove(sbColumnVaules.
                                   Length - 2, 2)
      sbSQL.Append(" (")
      sbSQL.Append(sbColumnVaules)
      sbSQL.Append(") ")
      sqlScripts.Add(sbSQL.ToString())
   Loop
Finally

   Reader.Close()
End Try

Return sqlScripts
End Function
```

La funzione restituisce una lista di stringhe, ciascuna delle quali è un'istruzione SQL. Osserviamo come funziona:

Viene eseguita la query che restituisce il contenuto della tabella. Si cicla, quindi, il DataReader con i valori e, per ogni riga, si costruisce la query.

```
sbSQL.Append("INSERT INTO [" & tableName & "] ")
```

fornisce la prima parte dell'insert.
Viene poi ciclata la lista delle colonne per scrivere i nomi dei campi:

```
For Each tc As TableColumn In tableColumns
    sbColumns.Append("[" & tc.COLUMN_NAME & "], ")
Next
```

Inseriamo, quindi, la scritta VALUES e poi i valori ciclando ancora sulle colonne:

```
For Each tc As TableColumn In tableColumns
sbColumnVaules.Append(FormatInput(Reader(tc.
                                   COLUMN_NAME), _
     tc.DATA_TYPE) + ", ")
Next
```

Nello scrivere i valori, utilizziamo la funzione ***FormatInput***, che formatta la stringa di input in base al tipo di dato della colonna.
Dopo aver generato lo script, la funzione ***SaveSQLScript***, si occupa di salvare su file i contenuti. Una particolarità della funzione in questione è data dalla seguente istruzione:

```
f.WriteLine("SET IDENTITY_INSERT [" & tableName &
                                         "] ON")
```

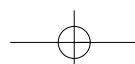
la quale dice a SQL Server che si possono inserire esplicitamente i valori dei campi identity.

CONCLUSIONI

Abbiamo visto come sia possibile ricavare in modo abbastanza semplice le informazioni sugli oggetti dei nostri database.
È meglio usare i SYSOBJECTS oppure le viste TABLES? Le informazioni che possiamo ottenere sono praticamente le stesse.
Le TABLES, rispetto ai SYSOBJECTS mostrano soltanto gli oggetti su cui l'utente ha i permessi; inoltre fanno parte di uno standard e sono quindi, a mio avviso, da preferire.
In ogni caso i metodi illustrati possono risultare molto utili nella migrazione dei dati quando le procedure standard non si atdattano alla nostra struttura, o quando la migrazione deve avvenire fra sistemi che hanno un diverso approccio alla gestione dei database.

Carmelo Scuderi

L'AUTORE

Carmelo Scuderi è ingegnere informatico. Si occupa di sviluppo software in ambiente .Net per una società di informatica di Milano. Gestisce un sito ricco di script e manuali per chi si affaccia al mondo della programmazione web (www.morpheusweb.it). le operazioni di boxing ed unboxing che erano invece indispensabili, ad esempio, con gli ArrayList, i cui elementi sono degli object.

APACHE DERBY: IL DATABASE TASCABILE

UNA PANORAMICA SUL DATABASE RELAZIONALE SCRITTO INTERAMENTE IN JAVA E RECENTEMENTE INTEGRATO DA SUN MICROSYSTEMS NELL'ULTIMA VERSIONE DEL JDK. MOLTO COMODO PER LE SUE RIDOTTE DIMENSIONI

Quando ci si trova a realizzare un progetto software di qualsiasi tipo, ci si pone spesso il quesito di come memorizzare i dati per poi successivamente recuperarli e/o elaborarli. Sviluppare personalmente una soluzione per risolvere questo problema risulterebbe molto dispendioso e richiederebbe conoscenze avanzate. Nella maggior parte dei casi questa incombenza viene affidata a prodotti di terze parti: i database relazionali. Scegliere una base dati conforme allo standard SQL *(Structured Query Language)* rappresenta un vantaggio notevole in quanto tale notazione, oltre ad essere riconosciuta universalmente come strumento di manipolazione dati, minimizza o annulla gli sforzi nel momento in cui si decida di cambiare RDBMS. Nello sviluppo Java, la tecnologia JDBC rende semplice ed intuitivo l'invio di comandi SQL ad un database relazionale in grado di comprendere tale "dialetto". Una delle novità presenti nell'ultima versione della Virtual Machine di Sun Microsystems è proprio l'integrazione con un database relazionale: Java DB. Non si tratta di un prodotto realizzato dal team di James Gosling ma di una ri-distribuzione, "marchiata Sun", del progetto open source Apache Derby. Nel prosieguo dell'articolo vedremo come utilizzarlo al meglio descrivendone le principali funzionalità.

APACHE DERBY

Le origini di Apache Derby risalgono al 1996, allora sotto il nome di JBMS. Successivamente fu ribattezzato Cloudscape e infine, dopo l'acquisizione da parte di IBM nel 2004, venne incluso nel progetto Derby di Apache. Fin dalle prime versioni, il database si dimostrò prodotto valido e funzionale, tanto da suscitare l'interesse di un colosso mondiale quale IBM. Negli anni successivi tante funzionalità furono potenziate e implementate, ma significative migliorie furono apportate quando il progetto diventò open source e solo grazie alla partecipazione della community che è stato possibile raggiungere il livello di stabilità e robustezza che oggi lo contraddistingue. Da un punto di vista tecnico è possibile tradurre questi dieci anni di sviluppo in un database engine completamente realizzato in Java eseguibile in due differenti modalità: *embedded* o *client-server*.

EMBEDDED O CLIENT-SERVER

La scelta della modalità da utilizzare è strettamente dipendente dalla tipologia di applicazione su cui ci troviamo a lavorare. Nel caso in cui l'applicazione faccia un uso esclusivo del database diventa preferibile utilizzare Derby in versione *embedded*: l'engine è eseguito all'interno della stessa JVM utilizzata dall'applica-

COME INIZIARE

É indispensabile avere installato sul computer Java 2 Standard Edition SDK 1.1 o superiore. É preferibile utilizzare l'ultima versione disponibile dal sito *http://java.sun.com*. **Scaricare lo zip file contenente il database Apache Derby dal sito** *http://db.apache.org /derby /derby_downloads.html*. **É possibile installare il database, indipendentemente dalla piattaforma sulla quale sarà utilizzato, estraendo il contenuto dell'archivio scaricato sul file system. La cartella *db-derby-<version>-bin* sarà creata automaticamente. Il "cuore" del sistema risiede nel file *derby.jar* contenuto nella directory *lib*. Per l'utilizzo all'interno di un determinato progetto è necessario dichiararlo nel CLASSPATH del progetto stesso. La documentazione risiede nella cartella docs ed è fruibile nei formati HTML e PDF. Una volta completata la fase di installazione del database è possibile eseguire lo script sysinfo per visualizzare le proprietà e l'ambiente in cui Derby verrà eseguito.**

zione. La caratteristica principale di questa opzione risiede nell'elevata velocità di esecuzione: i dati non hanno infatti bisogno di essere convertiti e spediti in rete come avviene nelle comunicazioni di tipo client-server. Ovviamente una sola applicazione (o, meglio, JVM) per volta può effettuare l'accesso a questo tipo di base dati (vedi **Figura 1**).

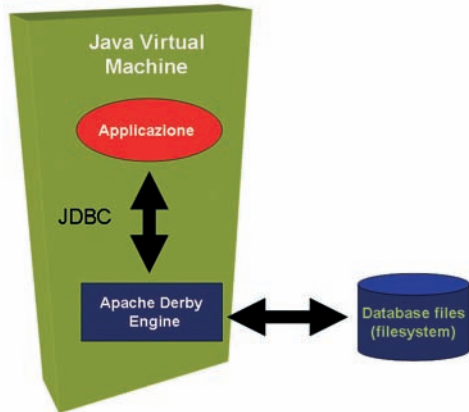



Fig. 1: Schema d'uso di Apache Derby in modalità embedded.

Di seguito è riportato il codice necessario ad ottenere una connessione ad un'istanza embedded:

```
// carico il driver di tipo embedded
try {
  Class.forName(
            "org.apache.derby.jdbc.EmbeddedDriver");
} catch (ClassNotFoundException cnfe)
{
    System.err.println("Unable to load jdbc driver:
                                " + cnfe.getMessage());
}
Connection conn = null;
try {// ottengo la connessione
    conn = DriverManager.getConnection(
            "jdbc:derby:ioProgrammoDB;create=true");
    SQLWarning warning = conn.getWarnings();
    if (warning != null) { // cerca eventuali avvertimenti
        System.out.println("WARNING: " + warning);
    }
}
catch (SQLException se)
{
    System.err.println("Unable to retrieve connection:
                                " + se.getMessage());
} finally {
    if (conn != null) {
        try {
            conn.close();
        } catch(SQLException se) { /* empty */ }
    }
}
```

L'URL standard per questa modalità è identificato dalla stringa:

```
jdbc:derby:[databaseName].
```

Inoltre, come riportato nel codice di esempio, è possibile inserire una serie di attributi separati dal delimitatore ";" (punto e virgola). Nel nostro caso, l'URL è stato completato con l'aggiunta del parametro create la cui funzione è quella di creare automaticamente la base se non esiste. In caso contrario viene assegnato alla connessione un oggetto di tipo *java.sql.SQLWarning* contenente un apposito avviso.
La modalità client-server (vedi **Figura 2**) viene generalmente utilizzata quando si vuole garantire un accesso multi utente sia locale sia remoto. L'interoperabilità con il database engine, che confronto alla precedente opzione rimane invariato, viene gestita da un server che utilizza il protocollo standard DRDA *(Distributed Relational Database Architecture)* attraverso cui è possibile comunicare con applicazioni scritte con differenti linguaggi (PHP, Perl, .NET).

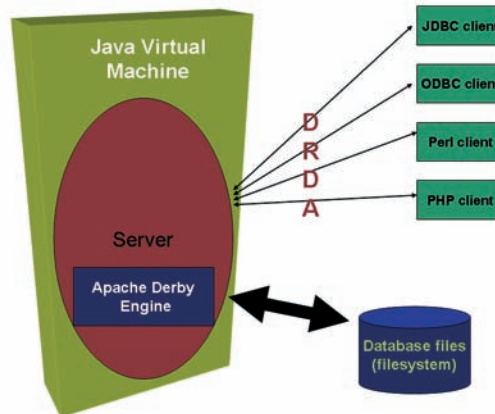
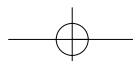



Fig. 2: Schema d'uso di Apache Derby in modalità client-server.

Non ci resta che avviare il database server. Per far questo è necessario eseguire lo script *startNetworkServer* presente nella directory di installazione di Derby all'interno del folder bin. Per creare una nuova base dati e stabilire una connessione remota alla stessa è possibile utilizzare lo stesso codice riportato per la modalità *embedded*, sostituendo la classe del driver con *org.apache.derby.jdbc.ClientDriver* e l'URL con *jdbc:derby://[hostName]:1527/ioProgrammoDB;create=true*, dove *hostName* rappresenta il nome o l'indirizzo IP della macchina su cui è stato avviato il server.
Di seguito riportiamo un metodo di esempio che, indipendentemente dalla modalità utilizzata, inserisce un record rappresentante un libro nella tabella *BOOKS* e ne stampa a video l'intero contento:

PRINCIPALI CARATTERISTICHE DI APACHE DERBY

- **Scritto interamente in Java**
- **Dimensioni ridotte, buone prestazioni ed affidabilità**
- **Nessun limite applicativo sulle dimensioni del database**
- **Facile da usare, non richiede amministrazione**
- **Supporto per triggers e Java stored procedures**
- **Supporto transazionale di tipo ACID *(Atomicity, Consistency, Isolation, Durability)***
- **Indipendente dalla piattaforma**
- **Supporta gli standard *J2ME/CDC/Foundation Profile* (JSR-169)**
- **Utilizzabile in modalità embedded o client-server**
- **La portabilità è garantita**
- **Compatibile agli standard ANSI-92 SQL e JDBC 3.0**
- **Il codice sorgente è disponibile**

```
public void execute(int id, String title, String author,
                                                   int pages)
{
    Connection conn = this.getConnection();
    if (conn != null)
    {
        Statement stmt = null;
        boolean create = false;
        try {
            stmt = conn.createStatement();
            stmt.executeQuery("select * from BOOKS");
        } catch (SQLException se) {
            create = true;
        }
        try {
            if (create) {
                System.out.println("La libreria non esiste
                                              … creiamola");
                stmt.execute("create table BOOKS
                           (ID integer, TITLE varchar(256),
                 AUTHOR varchar(512), PAGES integer)");
            }
            stmt.execute("insert into BOOKS values
                               ("+id+", " +"'"+title+"', " +
                         "'"+author+"' ," + pages+")");
            System.out.println("La libreria contiene:");
            ResultSet rs = stmt.executeQuery("select *
                                             from BOOKS");
            while(rs.next()) {
                System.out.println();
                System.out.println("ID :
                                    " + rs.getObject("ID"));
                System.out.println("TITOLO :
                                  " + rs.getObject("TITLE"));
                System.out.println("AUTORE :
                                 " + rs.getObject("AUTHOR"));
                System.out.println("PAGINE :
                                  " + rs.getObject("PAGES"));
            }
        } catch(SQLException se) {
            se.printStackTrace();
        }
    }
}
```

APPLICAZIONI WEB: DAL SERVER AL CLIENT

Con il Web 2.0 e l'evoluzione delle tecnologie ad esso associate, prima tra tutte AJAX, le applicazioni web hanno preso la tendenza di trasferire una buona parte della logica applicativa dal server al client (il web browser). Una limitazione particolarmente sentita dagli utenti è il fatto che tali applicazioni web necessitano di una connessione internet sempre attiva. Una soluzione a questo problema è quella di affiancare l'applicazione con un database eseguito sul browser stesso. Già da qualche tempo Mozilla sta analizzando la possibilità di equipaggiare Firefox con un db SQLite. Le difficoltà di integrazione sono relative al pesante utilizzo di risorse necessarie ad eseguire un database engine. Grazie alle sue caratteristiche, Derby si candida come valida soluzione a questo problema: il db è contenuto in un unico file e, a tempo di esecuzione, consuma all'incirca 2MB di memoria. Per "trasportare" il database dal server al client è possibile utilizzare una comune applet:

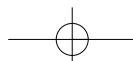
```
<APPLET CODE="DerbyLocalStoreApplet.class"
    CODEBASE="./ioProgrammo" WIDTH=1 HEIGHT=
                        1 NAME="DerbyLocalStore"
         ARCHIVE="derby.jar"></APPLET>
```

L'engine è contenuto nel file derby.jar mentre l'applet DerbyLocalStoreApplet si occuperà della sua gestione. Nel metodo init(), facente parte del ciclo di vita dell'applet, implementeremo la fase di inizializzazione del database:

```
public void init() {
    Connection conn = null;
    Statement stmt = null;

    System.out.println("Inizializzazione
                              del database locale");

    ds = new EmbeddedDataSource();
    // step di creazione del database
```

TRANSAZIONALITÀ IN PILLOLE

Quando si inizia l'analisi ed il disegno architetturale di un progetto software ci trova spesso a scegliere la base dati su cui si dovrà successivamente lavorare. Questa scelta è vincolata dalle caratteristiche implementate dai vari vendor. Un'importante funzionalità, soprattutto nei progetti su piattaforma enterprise, è il supporto alla transazionalità. È importante però sapere che esistono vari livelli di transazionalità riassumibili dall'acronimo ACID:

- **ATOMICITÀ: garantisce che una serie di operazioni contigue appaiano come una singola unità di lavoro.**
- **CONSISTENZA: garantisce che, alla fine di ogni transazione, lo stato del sistema sia consistente.**
- **ISOLAMENTO: permette l'esecuzione concorrente di più transazioni in modo del tutto trasparente alle varie transazioni coinvolte, garantendo che ogni transazione sia appunto isolata dalle altre.**
- **DURABILITÀ: garantisce che, in caso di malfunzionamenti (interruzione della comunicazione, crash del disco rigido), le risorse in fase di aggiornamento non vadano perdute.**

```
    ds.setCreateDatabase("create");
    ds.setDatabaseName("ioProgrammoDB");

    try {
        conn = ds.getConnection();

    } catch (SQLWarning sqle) {
        System.out.println("Il database
                    ioProgrammoDB è già presente ...");
        ds.setCreateDatabase(null);
    } catch (Exception ex) {
        System.err.println("ERRORE: Impossibile
            connettersi al database ioProgrammoDB: " +
                                    ex.getMessage());
        return;
    }

    // creazione della struttura del db

    try {
        stmt = conn.createStatement();
        stmt.execute("CREATE TABLE BOOKS
                    (ID integer, TITLE varchar(256)," +
                " AUTHOR varchar(512), PAGES integer)");
    } catch (SQLException e) {
        System.out.println("La tabella BOOKS
                                    già presente ...");
        try {
            stmt.close();
            conn.close();
            return;
        }catch (Exception ex) { /** empty */ }
    }
}
```

In questo frangente è stato utilizzato l'oggetto EmbeddedDataSource contenuto in Apache Derby. A fronte di particolari interazioni utente lato browser sarà sufficiente richiamare altri metodi, implementati all'interno della nostra applet, che andranno ad operare sulla base dati locale. Per ottimizzare il tempo di download dell'applet contenente la libreria derby.jar è consigliabile utilizzare pack200, un tool, contenuto nella JDK dalla versione 5.0, che consente di diminuire la dimensione dei Jar files (*http://java.sun.com/j2se/1.5.0/docs/guide/deployment/deployment-guide /pack200.html*).

RMS ADDIO?

In un precedente articolo di ioProgrammo abbiamo parlato del Record Management System, il sistema standard di persistenza dati presente su piattaforma J2ME con profilo MIDP 2.0. Tale tecnologia consente la memorizzazione e la successiva lettura di record in formato binario. Le funzionalità di questo sistema sono relativamente "grezze", pertanto viene solitamente utilizzato per applicazioni di piccole dimensioni.
Molti dispositivi di nuova generazione, equipaggiati con configurazione CDC (Connected Device Configuration), saranno in grado di eseguire l'API JDBC standard senza sacrificarne alcuna funzionalità. Viste le sue caratteristiche di database "tascabile" Apache Derby potrà essere utilizzato come base dati locale anche su tali dispositivi.

CONCLUSIONI

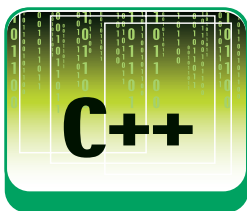
Apache Derby è un database relazionale i cui primi sviluppi risalgono all'anno 1996. Da allora la community open source ha apportato importanti migliorie sia in termini di funzionalità che di robustezza. Gli ottimi risultati di questi anni di lavoro sono stati ulteriormente confermati dalle scelte fatte da IBM e Sun Microsystems di includere il database nei prodotti Cloudscape e Java DB. La maggior parte delle caratteristiche che lo contraddistinguono fanno si che esso venga considerato nello sviluppo di varie di tipologie di progetto.
I punti di forza che hanno accresciuto la sua popolarità risiedono nelle ridotte dimensioni e nel limitato utilizzo di risorse a tempo di esecuzione.
Queste evoluzioni rappresentano una vera e propria rivoluzione per gli sviluppatori, in quanto consentono lo sviluppo di applicazioni con basi di dati locali su dispositivi mobili e browser web.

Fabrizio Fortino

NETBEANS + DERBY

La maggior parte degli ambienti di sviluppo presenti sul mercato offrono funzionalità per interfacciarsi in maniera visuale con un database generico tramite la tecnologia *JDBC/ODBC*. Visti gli stretti "legami" tra la community di Netbeans e Sun Microsystems, l'ambiente di sviluppo è stato corredato (dalla versione 5.5) con un'apposita feature per la gestione di Derby/JavaDB. Nel CD allegato è disponibile un tutorial interattivo che descrive l'integrazione dei due prodotti, accessibile dal file *flash_tutorial/derby +netbeans.html*.

IL C++ INCONTRA LE REGULAR EXPRESSION

NON SI PUÒ SERIAMENTE PARLARE DI TEXT PROCESSING, SENZA TRATTARE ANCHE LE ESPRESSIONI REGOLARI. È VERO CHE IL C++ NON SUPPORTA LE REGEX? E QUALI SONO LE LIBRERIE CHE PERMETTONO DI SUPERARE IL PROBLEMA?

Eccoci al terzo appuntamento col mondo del text processing. Nei primi due abbiamo incominciato dalle fondamenta, analizzando come il C++ definisce internamente il concetto di carattere e di stringa; come è possibile concatenare e formattare stringhe usando gli stream standard, format e lexical_cast; e infine come manipolare le stringhe attraverso gli algoritmi standard e le estensioni di string_algo.
Oggi siamo in grado di spostarci ancora più avanti, parlando della validazione e delle stringhe. Con questo termine s'intende la verifica della loro adesione ad uno specifico modello formale. I programmatori si trovano comunemente a dover validare delle stringhe; come esempio basti pensare agli indirizzi e-mail, ai codici fiscali, e ai numeri di carta di credito.
Ne sappiamo certamente abbastanza per realizzare a mano tutte queste routine di controllo, usando le funzionalità che abbiamo imparato negli ultimi articoli, ma l'impresa può rivelarsi molto complessa: ci vuole poco a sconfinare nella creazione di veri e propri automi a stati finiti, difficili da scrivere, leggere e manutenere.
La maggior parte dei programmatori, invece, conosce già – almeno per sentito dire – una soluzione comoda, potente e versatile: le espressioni regolari (altrimenti dette regex). Ecco, ad esempio una semplice regex per la validazione di una carta di credito.

```
(\d{4}[-/\s]*){4}
```

Posso immaginare la vostra reazione. Se avete già avuto a che fare con le regex, la analizzerete per due secondi pensando "sì, può andare. Anche se forse può essere fatta meglio in questo modo...". Se non avete mai programmato con le regex, invece, questo semplice esempio può già sembrare totalmente incomprensibile. Non potendo sapere a quale categoria apparteniate, supporrò che non sappiate niente delle espressioni regolari.
Alla fine di quest'articolo, saprete costruire delle regex di media complessità. Capirete perfettamente l'esempio posto qui sopra (e molti altri), e saprete utilizzarlo praticamente nel vostro codice C++.

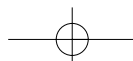
Conoscenze richieste
Buona conoscenza del C++

Software
Un compilatore C++ standard

Impegno

Tempo di realizzazione

IL C++ E LE REGEX

Le regex sono un argomento fortemente pratico. È molto facile scriverne una sbagliata, o che si comporta in maniera imprevista, soprattutto quando si sta ancora imparando – ma perfino gli esperti vanno spesso a tentativi. Sarebbe quindi auspicabile cominciare a spiegare le regex direttamente sul campo, con del codice C++ pronto per essere testato. Purtroppo, ciò presume che il C++ supporti in qualche modo le regex nativamente – e non è così. Lo standard non dice assolutamente nulla a riguardo, e questa è certamente una delle omissioni più gravi della libreria del C++. Chi volesse scrivere qualcosa di "standard" oggi potrebbe solo appoggiarsi alle funzioni POSIX regcomp, regexec, eccetera. Queste funzioni sono limitate, scomode, e di tipica impostazione C, qualcosa da evitare assolutamente nella programmazione C++.
Non è quindi una sorpresa che tanti programmatori abbiano cercato di colmare le lacune del C++ scrivendo una bella libreria che si occupasse di gestire le espressioni regolari. I tentativi sono stati molti, più o meno riusciti. Fra i migliori, non si possono non segnalare almeno Regex++ di John Maddock e GRETA di Eric Niebler. Regex++, in particolare, è stato adottato dalla comunità C++ più autorevole: boost ed è ad oggi una delle librerie più scaricate e usate dell'intera suite.
Il bisogno di un'implementazione standard per le regex in C++ ha portato alla stesura di un proposal nel TR1, basato proprio su regex++. Questo significa, in parole povere, che il C++ del futuro (C++09) avrà con tutta probabilità un supporto standard per le regex la cui interfaccia sarà molto affine a quella già fornita da boost::regex.
Una volta esaurite le premesse possiamo cominciare installando gli strumenti che ci servono. Le regex possono sembrare leggermente complesse ma tutto diventa semplice dopo il primo approccio.

INSTALLARE BOOST::REGEX

Per tutti i motivi esposti nei paragrafi precedenti, presenterò boost::regex come la soluzione C++ alle regex. Se volete seguire l'articolo, quindi, dovrete installare la libreria e configurare l'ambiente e gli eventuali IDE. In tutti i nostri precedenti appuntamenti boost ci ha sempre trattato molto bene, richiedendoci l'unico sforzo di #includere degli header.
Le regex, purtroppo, sono un caso particolare in cui oltre alle inclusioni dei file header è necessario compilare i sorgenti in una libreria. Fino a qualche tempo fa, boost era nota per non essere troppo amichevole in fase di installazione; oggi, per fortuna, le cose vanno diversamente. Nei passi che seguono vedremo che installare boost::regex è un'impresa relativamente semplice sia in Linux che in Windows. Per sapere dove reperire librerie e componenti, fate riferimento ai box laterali.

1 Scaricate **la libreria boost**, e mettetela in una cartella temporanea (come "C:\tmpBoost", o "/tmpBoost").

2 Scaricate boost-jam.

Per Linux, questa viene fornita come pacchetto rpm precompilato: vi consiglio di approfittarne.
Per Windows, scaricate la versione *ntx86.zip* di boost-jam e decomprimetela in *C:\tmpBoost*.

3 Se siete sotto Windows, assicuratevi di avere installato Visual Studio 2005 Express (IDE ottimo e gratuito), e il Platform SDK, e compiuto tutti i passi necessari alla loro configurazione (vedi box laterale).

4 Accedete al prompt dei comandi e portatevi nella directory di boost.

Per linux, la cosa è banale: entrate in un terminale qualsiasi, e digitate cd /tmpBoost.
Per windows, assicuratevi di accedere al terminale fornito dal Platform SDK sul menu d'avvio, che permette di avere tutte le variabili d'ambiente correttamente configurate; digitate "cd C:\tmpBoost".

5 Compilate.

Per Linux, inviate un comando sulla falsariga di:

```
bjam "-sTOOLS=gcc" install --prefix=/usr
```

e aspettate che termini la compilazione.

Per Windows, inviate un comando sulla falsariga di:

```
bjam "-sTOOLS=vc-8_0" install
```

e fatevi una lunga passeggiata, in attesa che termini la compilazione.

6 Impostate le librerie nell'IDE

Per Linux, impostate il vostro IDE in modo da passare al compilatore (anche) la direttiva **-lboost_regex-gcc-1_33_1.**
Per Windows, aggiungete alle directory di inclusione del vostro IDE "C:\Boost\Include\boost_1_33_1\" e a quelle di libreria "C:\Boost\Lib".

7 Cancellate la cartella temporanea (rimuovendo /tmpBoost o C:\tmpBoost)

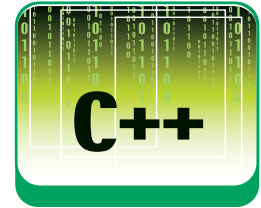
Complimenti! Avete ora un'installazione completa di boost a portata di mano (in /usr/ o in C:\Boost), che vi permetterà di usare non solo le regex, ma anche ogni altra meraviglia fornita da boost! Qui ho illustrato due percorsi possibili e basilari. Se doveste avere problemi o necessità particolari, consultate i riferimenti forniti sui box laterali per l'installazione con boost-jam o via makefile.

BOOST::BASIC_REGEX

Ora che abbiamo installato boost::regex possiamo fare la nostra prima prova:

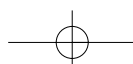
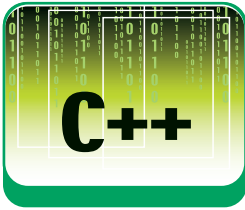
```
#include <iostream>
#include <boost/regex.hpp>

using namespace std;
using namespace boost;

int main()
{
  regex r("ciao!");
  cout << regex_match("ciao!", r);

  return 0;
}
```

Questo programma stamperà 1, e ora vedremo perché. Innanzitutto notiamo l'inclusione dell'header *<boost/regex.hpp>*, e della direttiva using namespace boost, che ci evita di dover ogni volta ricorrere al prefisso *boost::*.
Quindi, entriamo nel vivo con la riga *regex r("ciao!")*. Stiamo costruendo un oggetto r, di tipo regex. **boost::regex** è un typedef definito così:

```
typedef boost::basic_regex<char> boost::regex;
```

BOOST.ORG

Boost è "la" comunità C++ per quanto riguarda le estensioni alla libreria standard. L'indirizzo è www.boost.org, da dove si può scaricare liberamente l'intera libreria.

BOOST::REGEX

Le informazioni "ufficiali" su boost::regex si trovano all'indirizzo http://www.boost.org/libs/regex/doc/index.html.

Se avete seguito lo scorso appuntamento, questo non vi suonerà nuovo: anche la classe string è definita in modo molto simile:

```
typedef std::basic_string<char> std::string;
```

Regex è in effetti molto simile a string, perché ricalca l'interfaccia dei contenitori standard. Esistono regex e wregex, proprio come esistono string e wstring.
La stringa passata nel costruttore di boost::regex (in questo caso, "ciao!") è la regex vera e propria. Una volta definita, è possibile ottenere l'oggetto corrispondente come std::string, attraverso la funzione str():

```
regex r("ciao!");
string s(r.str());
```

Ovviamente, una regex non ha molto senso come contenitore alternativo a string fine a sé stesso. Lo scopo di una regex è fungere da modello in base al quale, poi, si effettueranno delle validazioni: nell'esempio ciò viene svolto dall'algoritmo **regex_match(testo_da_validare, regex)**.

BOOST-JAM

La maniera migliore per costruire le librerie di Boost è Boost-Jam: un sistema di makefile cross-platform. L'indirizzo è: http://sourceforge.net/project/showfiles.php?group_id=7586&package_id=72941.

CASE SENSITIVITY

Nel caso più banale, una stringa viene considerata valida rispetto ad una regex solo se è identica ad essa. Ciò spiega perché nel nostro codice di esempio regex_match restituisce true, e quindi viene stampato a schermo il valore "1". Da questo punto di vista, la chiamata a regex_match potrebbe essere sostituita da un semplice confronto con un oggetto string:

```
cout << (string("ciao!") == r.str()); // 1
```

Proprio come gli oggetti string, le regex sono case-sensitive, e pertanto se nel nostro esempio ci limitiamo anche solo a cambiare una maiuscola, regex_match ci restituirà inesorabilmente false:

```
regex r("ciao!");
regex_match("Ciao!", r); // false
```

Una simile distinzione non è sempre desiderabile. Qui comincia a sorgere una prima differenza con gli oggetti string: il costruttore di regex prevede un secondo parametro, all'interno del quale è possibile combinare dei flag che permettono di modificare i requisiti della regex. Uno dei più importanti è icase, che rende i confronti case insensitive.

```
regex r("ciao!", regex::icase); //regex case
                                          insensitive
regex_match("ciao!", r); //true
regex_match("CIAO!", r); //sempre true!!
```

CLASSI DI CARATTERI PREDEFINITE

Finora le regex non sono state granché esaltanti. Per fare un primo salto di qualità dobbiamo cominciare a fare distinzione fra varie "classi". Una "classe" è nel gergo delle regex un insieme di caratteri, come ad esempio quello delle cifre, dei caratteri maiuscoli, degli spazi, dei caratteri alfanumerici. Le classi più importanti sono indicate in tabella 1, sia nella loro versione posix "estesa" che nelle rispettive abbreviazioni con i backslash.

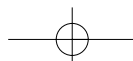
Significato	Classe	Breve
Qualsiasi carattere (tranne i ritorni a capo)	[^\r\n]	.
Cifre	[[:digit:]] o [0-9]	\d
Qualsiasi carattere tranne le cifre	[^[:digit:]] o [^0-9]	\D
Lettere Maiuscole	[[:upper:]] o [A-Z]	\u
Qualsiasi carattere tranne le maiuscole	[^[:upper:]] o [^A-Z]	\U
Lettere Minuscole	[[:lower:]] o [a-z]	\l
Qualsiasi carattere tranne le minuscole	[^[:lower:]] o [^a-z]	\L
Spazio	[[:space:]] o [\t\n\r\f\v]	\s
Qualsiasi carattere tranne gli spazi	[^[:space:]] o [^ \s]	\S
Carattere di parola (alfanumerici + underscore)	[[:word:]] o [\u\l\d_]	\w
Qualsiasi carattere tranne uno di parola	[^[:word:]] o [^\u\l\d_]	\W

Tabella 1: Classi predefinite di caratteri.

Per indicare l'ingresso di un numero a quattro cifre, ad esempio, si può scrivere la regex:

```
\d\d\d\d
```

Ed ecco il corrispondente programma per la validazione:

```
int main()
{
  regex r("\\d\\d\\d\\d");
  cout << regex_match("1234", r);  //true
  cout << regex_match("123", r);   //false!

  return 0;
}
```

È importante che notiate il **doppio backslash** all'inizio dei quattro "\\d". Ciò è una logica conseguenza del fatto che in C++ il backslash è un carattere speciale, che serve a introdurre i caratteri di escape (come il famosissimo \n, per andare a capo), e che per essere scritto testualmente dev'essere a sua volta preceduto da un backslash (traduzione: per scrivere \ in una stringa C++, bisogna scrivere \\).
A tanta confusione si aggiunge anche il fatto che pure le regex hanno i loro caratteri riservati, chiamati **metacaratteri**, e cioè .[{()*+?|^$.
Prendetevi un po' di tempo per capire le implicazioni di tutto questo: se volete scrivere quattro cifre seguite da un punto, ad esempio, dovrete scrivere questa regex:

```
\d\d\d\d\.
```

che corrisponde, però, a questo codice C++, con i doppi backslash:

```
regex r("\\d\\d\\d\\d\\.");
```

Potete verificare se avete capito bene quanto visto finora, analizzando questa regex:

```
regex r("\\d\\\\\\d");
```

Che tipo di espressione può essere validata da questa regex? La stringa 2\3 è valida? Perché?

CLASSI DI CARATTERI COMPOSTE

Oltre alle classi di carattere predefinite, è possibile indicare qualunque insieme di caratteri attraverso l'uso delle parentesi quadre. La regex:

```
\d[abc]\d
```

ad esempio, valida qualsiasi stringa costituita da una cifra, un carattere a scelta fra 'a', 'b' e 'c', e un'altra cifra. Ad esempio, la stringa: "1a2" è senz'altro valida. La stringa 2d1 non lo è, perché 'd' non fa parte della classe [abc].
Per snellire la specifica delle classi, spesso è comodo fare riferimento ai range. Poniamo di sviluppare un engine scacchistico, e di voler validare una coppia di coordinate della scacchiera. Vogliamo permettere stringhe come "a1", "b8", "h5", ma non "x9", o altri codici senza senso. Con le classi composte possiamo scrivere:

```
[abcdefgh][12345678]
```

ma è una scrittura un po' prolissa. Usando dei range con la sintassi [inizio-fine], possiamo ridurre il tutto a:

```
[a-h][1-8]
```

Le due scritture sono del tutto equivalenti.
Infine, come si può vedere dalla stessa tabella 1, può essere molto utile usare l'operatore di negazione '^'. Se si vuole evitare che l'utente inserisca un carattere non-italiano, ad esempio, si può usare il gruppo:

```
[^jkwxy]
```

QUANTIFICATORI

Le regex usate finora pongono un po' di problemi, soprattutto per il fatto che sembrano orientate ai singoli caratteri più che alle stringhe: ipotizziamo di dover validare un codice di 16 cifre, come quello di una comune carta di credito. Potremmo scrivere qualcosa del genere:

```
\d\d\d\d\d\d\d\d\d\d\d\d\d\d\d\d
```

Ma è davvero possibile che non esista la notazione ripeti per sedici volte? C'è, ovviamente, ed è questa:

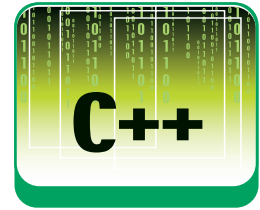
```
\d{16}
```

Le due scritture sono assolutamente equivalenti. {n} è un quantificatore che convalida la stringa solo se l'espressione che lo precede si trova ripetuta esattamente n volte. Dalla tabella 2, che mostra i principali quantificatori, scopriamo anche che possiamo scrivere il codice così:

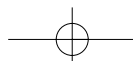
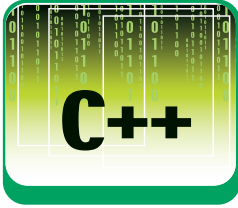
```
\d{15, 16}
```

Questa scrittura permette di trattare anche quelle carte di credito che hanno quindici cifre, e non solo quelle che ne hanno sedici. Il quantificatore {n, m}, infatti, valida la stringa solo se l'espressione che lo precede è ripetuto almeno n volte, e al più m volte. Sempre dalla tabella 2, scopriamo anche un'altra notazione sorprendente:

```
\w{6, }
```

Potremmo usare questa regex per chiedere all'utente di inserire una password di almeno 6 caratteri. Il quantificatore {n, }, infatti, valida la stringa solo se l'espressione che lo precede è scritta almeno n volte. Questa scrittura è ancor più potente di quanto appare, e fa fare alle nostre regex un bel salto di livello. È talmente utile che ha dato origine ai sinonimi *, +, e ?, mostrati nel resto della in tabella

NOTA

VISUAL C++ EXPRESS E PLATFORM SDK

Su Windows consiglio caldamente l'installazione di Visual C++ Express 2005 in congiunzione col Platform SDK. Per assicurarvi di installarli (e, soprattutto, configurarli) correttamente, seguite i semplici passi descritti a quest'indirizzo: http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/

2, che si traducono rispettivamente in {0, }, {1, } e {0, 1}. Ecco l'esempio della carta di credito, riveduto e corretto alla luce dei quantificatori:

```
regex r("\\d{4}[-/\\s]*\\d{4}[-/\\s]*\\d{4}[-
                              /\\s]*\\d{4}[-/\\s]*");

cout << regex_match("1234567890123456", r);
//true
cout << regex_match("1234-5678-9012-3456", r);
                                           //true
cout << regex_match("1234  - 5678 / 9012 /-/
                            3456", r);  //true!
cout << regex_match("123  - 5678 - 9012 - 3451",
r);
                                          //false!
```

Mi rendo conto che l'impatto con la regex alla prima riga potrebbe essere traumatico; non spaventatevi. Fate come fanno tutti gli esperti quando si trovano davanti ad un'espressione regolare: fate la faccia di quelli che: "sì, sì, è chiarissimo" (salvare le apparenze è importante), e intanto scomponetela mentalmente pezzo per pezzo. Diventerà facile sul serio. Togliete prima i doppi backslash, riducendoli ad uno solo.

```
\d{4}[-/\s]*\d{4}[-/\s]*\d{4}[-/\s]*\d{4}[-\s/]*
```

Quindi analizzate la regex con calma: si richiedono quattro numeri, seguiti da un carattere qualsiasi fra: - (il trattino), / (la slash), e \s (ovvero tutti i caratteri di spazio). Questo insieme di caratteri è seguito dall'asterisco, che vuol dire (tabella 2 alla mano) che può essere ripetuto zero o più volte. Poi il tutto si ripete uguale per altre tre volte. Questa regex serve quindi a validare un insieme di quattro numeri, seguito da un insieme di separatori (/ -), per quattro volte. Ora potete capire molto meglio perché i primi tre match funzionano, e l'ultimo invece no.

Quantificatore	Sintassi
Esattamente n volte	{n}
Da n a m volte (estremi compresi)	{n, m}
n o più volte	{n, }
zero o più volte ({0, })	*
una o più volte ({1, })	+
zero o una volta ({0, 1})	?

Tabella 2: Quantificatori.

GRUPPI

L'ultima regex è un po' lunga. La prima soluzione a cui verrebbe spontaneo pensare, potrebbe essere qualcosa del genere

```
\d{4}[-/\s]*{4}
```

L'idea è buona: ripetere quattro volte il gruppo "cifre più separatori". Peccato che non funziona: il quantificatore si applica solo all'ultimo elemento - quindi, in realtà stiamo "moltiplicando per quattro" soltanto [-/\s]*. Il che è sbagliato, e inutile. Per ripetere anche le cifre insieme ai separatori, dobbiamo usare le parentesi tonde, così:

```
(\d{4}[-/\s]*){4}
```

Ora il quantificatore si applica a tutto il "blocco" (\d{4}[-/\s]*). Un'espressione inclusa fra parentesi tonde viene chiamata in gergo gruppo. I gruppi, come vedremo, giocano un ruolo fondamentale per le ricerche e i riferimenti.

RICERCHE

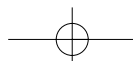
Finora abbiamo limitato la nostra attenzione alla funzione regex_match. Questa, come abbiamo visto, restituisce true solo se si verifica una corrispondenza perfetta fra la regex e la stringa da analizzare. Ciò è utilissimo nei casi in cui si voglia validare un testo, ma una regex può servire anche ad altro. L'algoritmo regex_search, ad esempio, può essere molto utile per implementare facilmente delle ricerche testuali con tutta la potenza degli strumenti più avanzati, come grep. Regex_search si comporta in maniera un po' diversa da regex_match: non richede una corrispondenza perfetta, ma la cerca in mezzo alla stringa passata. Esempio:

```
regex r("(\\d{4}[-/\\s]*){4}");
cout << regex_match("000-1234567890123456",
r);
                                          //false
cout << regex_search("000-1234567890123456",
r);
                                           //true
```

La regex della carta di credito viene ora utilizzata sulla stringa "000-1234567890123456". Questa non corrisponde perfettamente alla regex, in quanto contiene tre caratteri iniziali, e non quattro come richiesto. Pertanto, regex_match svolge onestamente il suo lavoro restituendo false. Regex_search, invece, fa un passo in più, e cerca di andare avanti nella speranza di trovare prima o poi una corrispondenza. E ci riesce, perché dal quinto carattere in poi ci sono sedici cifre, proprio come richiesto dalla regex. Regex_search è quindi felice di restituire true! Questo sistema apre nuovi scena-

NOTA

BIBLIOGRAFIA

Una buona introduzione a boost::regex (e non solo) è fornita in "Beyond the C++ Standard Library, an introduction to boost" di Bjorn Karlsson. L'ottimo "The C++ Standard Library Extensions: A Tutorial and Reference" di Pete Becker dedica molti capitoli alle regex così come sono standardizzate nel TR1. L'intero testo è', in generale, un validissimo punto di riferimento per capire i vari proposal del TR1 – che letti da soli sono molto sterili e poco accessibili. Per chi volesse approfondire l'argomento regex consiglio: "Mastering Regular Expressions" (seconda edizione) di Jeffrey E F Friedl. </BOX>

ri, nuove possibilità, (e, perché no, anche nuovi problemi) alle nostre regex.

RICERCHE E RISULTATI

Ora immaginiamo di essere degli odiosissimi spammer, sempre alla ricerca di nuovi strumenti per carpire gli indirizzi e-mail degli ignari utenti che ancora si ostinano a scriverli sulle pagine web. Probabilmente la nostra “stringa” consisterà in un'intera pagina HTML potenzialmente piena di indirizzi promettenti. Organizziamo una piccola regex senza pretese, per individuarli:

```
[\w\.]+@[\w\.]+\.\w+
```

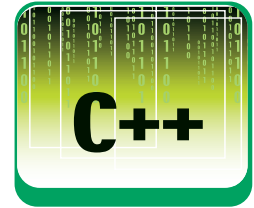
Il senso dovrebbe essere chiaro: “una parola (di almeno un carattere o punto)”, seguita da una chiocciola, seguita da un'altra parola (di almeno un carattere/punto), seguita da un punto, seguita da una parola (di almeno un carattere). Non è molto accurata, ma non dimenticate chi siamo e quali sono i nostri squallidi obiettivi. A questo punto proviamola su una semplice paginetta:

```
regex r("[\\w\\.]+@[\\w\\.]+\\.\\w+");
string pagina("per informazioni scrivete a
    posta@robertoallegra.it... l'email di ioprogrammo è
                              ioprogrammo@edmaster.it");

cout << regex_search(pagina, r); //true!
```

Ok, regex_search restituisce true. Ma non ci serve a molto sapere se una pagina contiene o meno degli indirizzi email. Vogliamo sapere quali sono. Per questo, dobbiamo richiamare un overload di regex_search che ci permetta di aver accesso al risultato della ricerca:

```
//...
smatch result;

if (regex_search(pagina, result, r))
  cout << result; //posta@robertoallegra.it
```

Ora regex_search stampa la nostra prima vittima: “posta@robertoallegra.it”. Tutto grazie all'oggetto result, di tipo smatch, che serve a contenere i risultati di una ricerca effettuata su un oggetto std::string (pagina). Smatch è, infatti, un typedef della classe generica match_results:

```
template <class Iterator,
  class Allocator=std::allocator<sub_match<Iterator>
>
    class match_results;

typedef match_results<const wchar_t> wcmatch;
typedef match_results<const char*> cmatch;
typedef match_results<std::string::const_iterator>
                                             smatch;
typedef match_results<std::wstring::const_iterator>
                                            wsmatch;
```

Come vedete, in questo caso abbiamo usato smatch, perché abbiamo dichiarato pagina come std::string. Se l'avessimo dichiarato come const char*, avremmo avuto bisogno di un cmatch, e così via. La funzione regex_search che abbiamo richiamato è diversa dall'ultima volta; ha tre argomenti: il testo su cui effettuare la ricerca, l'oggetto in cui immagazzinare il risultato, e infine la regex.

RISULTATI MULTIPLI

Nel nostro esempio, però c'è un secondo indirizzo email che ci è sfuggito: ioprogrammo@edmaster.it. Evidentemente regex_search si ferma al primo risultato trovato. Per accedere anche ai successivi dobbiamo utilizzare un ciclo, e ancora un nuovo overload di regex_search:

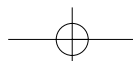
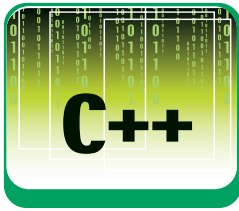
```
//...

string::const_iterator it  = pagina.begin();
string::const_iterator end = pagina.end();

while (regex_search(it, end, result, r)) {
  cout << result << endl;
  it = result.suffix().first;
}
```

Ora il programma stampa entrambi gli indirizzi, uno dopo l'altro. È importante capire quel che sta succedendo. Abbiamo usato un overload di regex_search che prende non più un oggetto string, ma due iteratori, proprio come fanno solitamente gli algoritmi standard. Ogni volta che viene stampato un risultato, si incrementa l'iteratore it, puntandolo al primo carattere (first) di ciò che sta dopo il testo trovato (suffix()). Il ciclo continua, finché non vengono più trovate corrispondenze.

RISULTATI E GRUPPI

Ora sappiamo inserire gli indirizzi trovati in una pagina HTML nel nostro database di vittime. Possiamo fare di peggio? Certamente! Sarebbe bello scomporre gli indirizzi in modo tale da separarne le componenti fondamentali. Possiamo usare i gruppi, così:

```
([\w\.]+)(@)([\w\.]+)\.(\w+)
```

Così facendo abbiamo scomposto l'indirizzo in

quattro gruppi: username (1), chiocciola (2), dominio (3), e dominio di primo livello (4).

```
regex r("([\\w\\.]+)(@)([\\w\\.]+)\\.(\\w+)");
//...

while (regex_search(it, end, result, r)) {
  cout << "indirizzo:\t" << result[0] << endl
       << "username:\t" << result[1] << endl
       << "chiocciola:\t" << result[2] << endl
       << "dominio:\t" << result[3] << endl
       << "primolivello:\t" << result[4] << endl
       << endl << endl;
  it = result.suffix().first;
}
```

Come vedete dal codice qui sopra, gli oggetti di tipo match_results sono utili anche come contenitori. Result[0] contiene l'intera corrispondenza trovata, e gli elementi dall'1 in poi corrispondono ai singoli sottogruppi appena descritti.

```
/bin/sh - Konsole
Sessione  Modifica  Visualizza  Segnalibri
Impostazioni  Aiuto

indirizzo:      posta@robertoallegra.it
username:       posta
chiocciola:     @
dominio:        robertoallegra
primolivello:   it


indirizzo:      ioprogrammo@edmaster.it
username:       ioprogrammo
chiocciola:     @
dominio:        edmaster
primolivello:   it


Press Enter to continue!

/bin/sh
```

Figura 1: Scomposizione del testo in sottogruppi mediante regex_search e match_results

L'AUTORE

Il sito www.robertoallegra.it contiene l'elenco degli articoli pubblicati in questa rubrica, con gli inevitabili approfondimenti ed errata corrige. L'e-mail dell'autore è articoli@robertoallegra.it

ALTERNATIVE

Inspiegabilmente, alcuni utenti hanno incominciato a infastidirsi per la nostra onesta attività di spammer, e hanno escogitato una contromisura. Non scriveranno più il proprio indirizzo come "username@dominio.com", ma lo altereranno leggermente, in modo che sia comprensibile per gli umani, ma non per la nostra regex. Scriveranno, ad esempio "usernameATdominio.com", oppure "username at dominio.com", eccetera. Che illusi! Abbiamo previsto la loro reazione, per questo abbiamo creato un gruppo fatto apposta per la chiocciolina. Ci basta modificarlo, così:

```
regex r("([\\w\\.]+)(@|at| at )([\\w\\.]+)\\.(\\w+)",
                                          regex::icase);
//...
```

Il gruppo della chiocciola, ora, comprende più alternative, separate dall'operatore pipe (|), che si comporta come un or logico. In result[2] potremo avere "@", "at", o " at ", a seconda dei casi.

REGEX_REPLACE

L'ultima modifica ci crea un problema: sappiamo leggere un indirizzo con delle chioccioline finte ("at", " at "), ma non potremo mai usarlo per spedire la nostra spazzatura. Prima di memorizzarlo nel nostro database dovremo quindi sostituire il secondo gruppo con una chiocciolina.
Possiamo realizzare la cosa concatenando le stringhe che abbiamo trovato, in questo modo:

```
string indirizzo = result[1] + "@" + result[3] + "." +
                                              result[4];
```

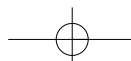
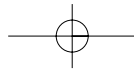
Questo risolve il problema: anche se result sarà "usernameATdominio.com", indirizzo conterrà comunque "username@dominio.com". Ma si tratta di una soluzione prolissa, un po' analoga alla concatenazione via-stream che abbiamo visto nel primo articolo di questa serie. Grazie all'algoritmo regex_replace, possiamo unire i risultati in un testo di formattazione, proprio come si fa nella creazione di stringhe con boost::format o sprintf:

```
string indirizzo(result[0]);
indirizzo = regex_replace(indirizzo, r, "$1@$3.$4"));
```

Regex_replace prende in ingresso un testo da modificare, una regex di partenza, e un testo di formato. Nel testo di formato ci si può riferire ai singoli sottogruppi con l'operatore $, permettendo così una scrittura estremamente compatta, e variabile.

CONCLUSIONI

In quest'articolo abbiamo solo grattato la superficie delle possibilità offerte dalle espressioni regolari e da boost::regex, e non potrebbe essere altrimenti, dal momento che una trattazione completa richiederebbe un libro intero (come quelli mostrati in bibliografia).
Il nostro viaggio nel text processing va avanti verso l'ultima puntata, in cui arriveremo a superare perfino le espressioni regolari, costruendo dei veri e propri parser. Non mancate!

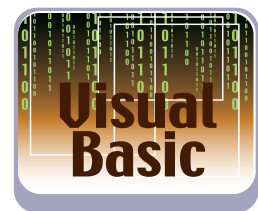
Roberto Allegra

UN GENERATORE DI CLASSI VISUAL BASIC

PROGETTIAMO UN GENERATORE AUTOMATICO A PARTIRE DA UNO SCHEMA XML. QUESTO GENERE DI APPLICAZIONE CI CONSENTE DI RISPARMIARE TEMPO. INOLTRE IMPAREREMO COME VB GESTISCE IL PARSINF DI FILE XML

Uno studio su un possibile utilizzo dello scripting, è lo scopo di questo articolo. Tutto ciò per realizzare un generatore automatico di classi VBScript che rimappino la struttura di un file XML qualunque, purché dotato di schema interno o esterno.

GLI SCHEMI XML

Una interessante caratteristica dell'XML è la possibilità di verificare la correttezza semantica di un certo file rispetto ad un template standard di riferimento detto schema. Infatti l'XML è di per sé un formato piuttosto rigido, nel senso di garanzia di correttezza sintattica che prevede una lunga serie di regole da rispettare, perché il file XML si possa definire well formed. Queste vanno dal chiudere ogni tag che viene aperto, all'uso degli apici per i valori degli attributi, alla presenza di un solo nodo padre per ciascun file e tanto altro ancora. Non sarà argomento di questo articolo spiegare queste regole; la rete e le librerie sono ormai sature di materiale sull'XML. La verifica sintattica però non è tutto. Si è reso necessario definire un formato che descrivesse le regole semantiche a cui deve soggiacere il file perché possa essere universalmente compreso anche da chi non ne è l'autore. Proviamo ad immaginare di voler definire un formato standard per le fatture da condividere con i nostri clienti e fornitori, e per far ciò vogliamo basarci su XML. Dobbiamo specificarne la logica (es. un nodo di testata che si chiama HEADER, che contiene alcuni attributi come Number, Year e altri ancora, uno o più nodi di tipo ROW e così a continuare). Lo schema XML serve proprio a questo. In realtà gli schemi sono un'invenzione recente per l'XML: inizialmente il loro compito era svolto dai DTD (Document Type Definition) dalla logica molto simile. Gli schemi hanno, rispetto ai loro antesignani, numerosi vantaggi: sono XML essi stessi, supportano l'ereditarietà e sono molto più potenti e completi. Osserviamo questo esempio di file XML:

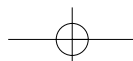
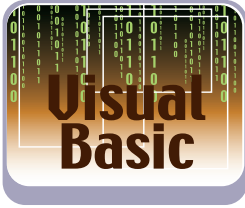
```
<?xml version="1.0"?>
<vCRM AUTHOR="Vito Vessia"
    mlns:xsi="http://www.w3.org/2001/XMLSchema-
                                        instance"
 xsi:noNamespaceSchemaLocation="F:\ScriptClasses\
                    XMLClasses\disk\Untitled2.xsd">
  <ORDER DONE_NODE="FALSE"
                          CODE_SITE="JRHOLD"
                 CODE_CORRESPONDENT="ARDILLO"
   CODE_INVOICE_CUST="029900"
                      CODE_SHIP_CUST="029900"
 CODE_ORDER_TYPE="%" CODE_FAIR="COD_FIERA"
   CODE_PL_SOFA="AJ" CODE_PL_ACCESS="EF"
                     CODE_CURRENCY="ITL[~]"
                        COD_CAUS_COMM="G03"
   COD_CAUS_PROD="MAG"
                        CODE_DOC_TYPE="003">
      <HEADER_REMARK
                    HRMRK_CODE_SITE="JRHOLD"
     HRMRK_PROGRES="1" HRMRK_CODE="OCMC"
      HRMRK_REMARK="Maximum[~]Care"
    HRMRK_ON_CREDITS="0" HRMRK_ON_DDT="1"
                          HRMRK_ON_LABEL="0"
      HRMRK_ON_INVOICE="1"
                            HRMRK_ON_OC="1"/>
           <ROW CODE_DBALT="031615"
    CODE_PL="AJ" ROW="1" NUM_REVIS_ROW="0"
                          ORDER_SET="2" PO=""
   CODE_PART="073504224">
             <ROW_REMARK
                     RRMRK_CODE_SITE="JRHOLD"
      RRMRK_PROGRES="2" RRMRK_CODE="OCMC"
RRMRK_REMARK="Maximum[~]Care"/>
           <COVER CODE_COVER="15001011"
 CODE_QUEST="Z0110001" FLG_ROW_MODIFIED=""
                                PROGR_RIV="1"
NUM_REVIS_COVER="0"
```

Conoscenze richieste

Visual Basic 6 livello intermedio

Software

Microsoft Windows 2000 o successivi e Microsoft Visual Basic 6.

Impegno

Tempo di realizzazione

```
                    CVR_CODE_SITE="JRHOLD"/>
        </ROW>
    </ORDER>
</vCRM>
```

In **Figura 2** è possibile osservare lo schema visualizzato all'interno di un editor multifunzionale. In **Figura 3**, invece, il browser mostra il file XML usato nelle prove e che soggiace allo schema precedentemente introdotto.

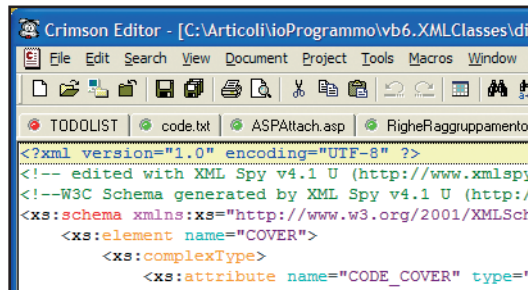



Fig. 1: Lo schema XML

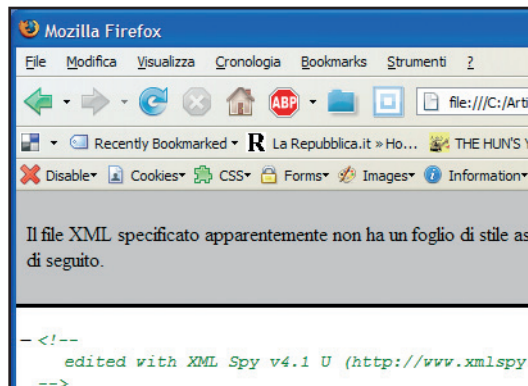



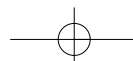
Fig. 1: Il file XML visto dal browser

Ecco di seguito lo schema di riferimento del file XML appena mostrato. In realtà l'esempio è stato privato di numerosi attributi e di alcuni nodi per ridurne l'occupazione tipografica, ma rende l'idea. Per osservare la versione integrale dei due file XML di dati e di schema d'esempio usati nell'articolo e per consultare il codice Visual Basic, si rimanda al sorgente che accompagna l'articolo.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
                elementFormDefault="qualified">
        <xs:element name="vCRM">
            <xs:complexType>
                <xs:sequence>
                    <xs:element
                        ref="ORDER"/>
                </xs:sequence>
                <xs:attribute
```

```
name="AUTHOR" type="xs:string" use="required"/>
                        <xs:attribute
    name="DATE" type="xs:string" use="required"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="ORDER">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element
    ref="HEADER_REMARK" maxOccurs="unbounded"/>
                            <xs:element
            ref="ROW" maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute
            name="DONE_NODE" type="xs:string"
                            use="required"/>
                        <xs:attribute
            name="CODE_SITE" type="xs:string"
                            use="required"/>
                    </xs:complexType>
                </xs:element>
</xs:schema>
```

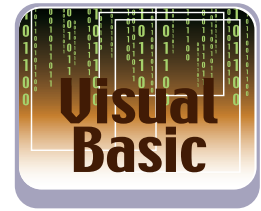
Possiamo notare come il file XML presenta un riferimento allo schema a cui soggiace (Untitled2.xsd) e che lo schema non è altro che un contenitore di tutti i tipi di nodi presenti nel file (nodi xs:element), con la struttura di ogni attributo che contiene (xs:attribute) compresi i tipi e le eventuali constraint (es. use=”required”). L'elenco dei nodi è flat, cioè non presenta la tipica struttura gerarchica dei file XML che lega i nodi padre ai nodi figlio: questa gerarchia è invece definita attraverso la presenza dei nodi xs:sequence definiti nei nodi xs:element.

Il principio di funzionamento è molto semplice: per verificare che il file XML sia semanticamente corretto rispetto allo schema di riferimento, deve soddisfare tutte le regole di quest'ultimo. Questa verifica di correttezza sarebbe un po' troppo onerosa, anche se non impossibile, se realizzata a mano affidandosi solo al nostro colpo d'occhio, ma in realtà il compito è realizzato egregiamente da un buon parser XML di ultima generazione. Si intende cioè un parser in grado di manipolare e di comprendere gli schema XML di ultima generazione definiti dal World Wide Web Consortium. Nel nostro esperimento utilizzeremo il potente parser Microsoft XML Core Services (MSXML) 4.0. Questo componente racchiude al suo interno un classico DOM (Document Object Model), cioè un oggetto in grado di produrre un completo object model a partire da un file XML, un SAX (Simple API for XML) cioè un componente che non conserva in memoria l'intera struttura del file

XML ma scatena degli eventi durante il parsing dello stesso. È certamente più performante del DOM anche se richiede una maggiore elaborazione e offre meno servizi. La vera novità della versione 4.0, rispetto alle precedenti, è invece la presenza del SOM (Schema Object Model): un componente in grado di produrre un object model a partire da uno schema XML, come esposto nel riquadro.
Data la natura completamente plain text del formato XML, sarebbe possibile editare, creare e modificare questi file e gli schemi associati direttamente col Notepad, ma ci si rende conto ben presto che questo ha del disumano. Ed è soprattutto vero quando si usano pesantemente gli schemi. Dunque è bene affidarsi ad un buon strumento specifico per questo tipo di operazioni. Visual Studio 2002 o successivi ne sono dotati, ma purtroppo non altrettanto il nostro buon vecchio Visual Basic 6, e così dovremo affidarci alle terze parti.

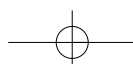
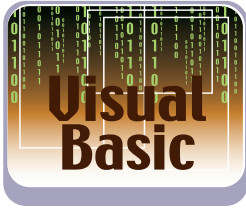
COSA C'ENTRA LO SCRIPTING?

Giunti a questo punto, vi chiederete cosa c'entra lo scripting in tutto questo. In effetti non c'entra apparentemente nulla se non fosse che si possono realizzare interessanti applicazioni dall'abbinamento delle tecnologia di scripting ActiveScript di Microsoft e l'XML con schema. Infatti il disaccoppiamento tra schema XML e dati XML ricorda da vicino la metafora OOP della classe e dell'oggetto: la classe è la definizione formale delle regole e dei comportamenti da seguire un particolare aggregato strutturale di dati, l'oggetto è invece la stessa classe che prende vita e va in esecuzione su un particolare set di dati che corrisponde al prototipo dell'aggregato detto in precedenza. Così se la definizione dei dati e dei comportamenti sugli stessi (la classe) è unica per una particolare tipologia applicativa, le sue "istanze" (gli oggetti) specifiche basate su set di dati differenti possono essere invece numerose e senza relazioni tra loro. Così, nel caso specifico, possiamo immaginare di produrre un set di classi, un intero object model, a partire da uno schema XML. Fatto questo saremo in grado di produrre istanze di queste classi che si basano su file XML differenti, purché soggiacciano allo schema comune da cui sono state generate le classi.
Vediamo un esempio di classi VBScript che permettono di navigare il file XML dati.xml presente nella directory dei sorgenti che accompagnano l'articolo:

```
'codice VBScript
Class vCRM
    Private m_Dictionary
    Private m_OnStartup
    Private m_VERSION
    Private m_AUTHOR

    Public CUSTOMER
    Public ORDER

    Private Sub Class_Initialize()
        set m_Dictionary =
                    CreateObject("Scripting.Dictionary")
        m_OnStartup = True
        set CUSTOMER =
                    CreateObject("Scripting.Dictionary")
        set ORDER =
                    CreateObject("Scripting.Dictionary")
    End Sub

    Private Sub Class_Terminate()
        set m_Dictionary = Nothing
        set CUSTOMER = Nothing
        set ORDER = Nothing
    End Sub

    Public Property Get Attributes()
        Set Attributes = m_Dictionary '.Items
    End Property

    Public Property Get AUTHOR()
       AUTHOR = m_AUTHOR
    End Property

    Public Property Let AUTHOR(valueField)
       m_AUTHOR = valueField
    End Property
End Class
```

Quella appena mostrata è la classe di navigazione del nodo vCRM del file XML in questione (file già parzialmente riportato in precedenza in questo articolo). Osservando le funzioni in essa contenute, possiamo notare la presenza di property let/get per accedere alla proprietà AUTHOR che, nel file XML, è l'attributo AUTHOR. Esisterà quindi una coppia di questi metodi nella classe per ogni attributo nel file XML d'origine. Possiamo inoltre notare la presenza della property get Attributes che rappresenta la collezione di tutti gli attributi all'interno della classe in modo che vi si possa accedere per indice e per chiave e, soprattutto, si possa applicare l'enumerazio-

NOTE

RIFERIMENTI

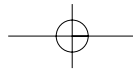
[1] Don Box, A. Skonnard, J. Lam - Essential XML – Beyond markup – Addison Wesley (2000)
[2] I tipi di dati di SOAP (revisione del 2001), http://www.w3.org/2001/XMLSchema
[3] XML Schema Part 0: Primer - http://www.w3.org/TR/xmlschema-0/
[4] Documentazione a corredo del MSXML 4.0 (xmlsdk.chm)
[5] Sezione dedicata allo scripting -, http://msdn.microsoft.com/script

ne COM (la for each).
Così, per accedere all'elemento AUTHOR potremmo invocare:

```
'codice VBScript
Dim oCRM

Set oCRM = New vCRM
oCRM.AUTHOR = "Vito Vessia"
Msgbox oCRM.AUTHOR

oppure accedere in modo simbolico alla proprietà:

MsgBox oCRM.Attributes("AUTHOR")

e scorrerle tutte semplicemente con:

'codice VBScript
Dim lAttribute

For Each lAttribute in oCRM.Attributes
    Msgbox lAttribute
Next
```

La collezione degli attributi è realizzata usando un Dictionary a livello di classe che viene istanziato nella Class_Initialize. Osservando però il nodo XML vCRM, possiamo notare che esso possiede alcuni nodi figli. Uno di questi è ORDER, la cui classe, secondo il prototipo appena visto, dovrebbe essere come quella che segue (per ragioni di praticità in realtà se ne riporta solo un estratto):

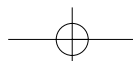
```
'codice VBScript
Class Order
    Private m_Dictionary
    Private m_OnStartup
    Private m_CODE_FAIR
    Private m_COD_CAUS_PROD
    Private m_TOT_SIT

    Public HEADER_REMARK
    Public ROW

    Public Property Get TOT_SIT()
      TOT_SIT = m_TOT_SIT
    End Property

    Public Property Let TOT_SIT(valueField)
      If Not IsNumeric(0 & valueField) Then
        Err.Raise 1970, , "Invalid number!"
      End If
      m_TOT_SIT = valueField
    End Property
End Class
```

Possiamo notare che la classe vCRM presentava una proprietà ORDER. Questa rappresenta proprio la collezione dei nodi figli che, a tutti gli effetti, sono classi di tipo Order come quella appena mostrata. Questa collezione è realizzata, ancora una volta, col solito Dictionary. Così, se vogliamo accedere e stampare a video la proprietà TOT_SIT del secondo nodo ORDER di una classe vCRM che fa da proxy al nostro XML vCRM, dobbiamo semplicemente scrivere:

```
Msgbox vCRM.ORDER(2).TOT_SIT
```

A sua volta, il nodo Order conterrà altri nodi figli ripercorrendo la stessa gerarchia del file XML originario, così da realizzare un modello ad oggetti. Nel caso specifico i nodi figli di Order sono HEADER_REMARK e ROW e così via fino a completare la gerarchia (es. vCRM.ORDER ( ). Si può vedere quindi come è possibile realizzare classi VBScript (o di scripting in generale in tecnologia Microsoft ActiveScripting) che ripercorra, in forma più naturale ed immediata, la struttura di un file XML qualsiasi basandosi sullo schema a cui soggiace il file stesso. Sarà così possibile istanziare queste classi con i dati presenti nel file XML, infatti ogni attributo del nodo XML è esposto nella classe in forma di property get/let che permetterà quindi anche di assegnare i valori di queste proprietà:

```
vCRM.ORDER(2).TOT_SIT = 3
```

Realizzare un'intera applicazione in VBScript è probabilmente poco pratico e ancor meno interessante dal punto di vista dell'implementazione in una situazione reale. Si dà il caso però che sia possibile utilizzare proficuamente tali classi in applicazioni Visual Basic (VBA) o in generale in tutti quei linguaggi ad alto livello capaci di gestire le interfacce IDispatch. Brevemente, vediamo come utilizzare direttamente da Visual Basic tali classi grazie al potente Microsoft Script Control **[5]** che è liberamente scaricabile dal sito di Microsoft e che semplifica drasticamente l'uso della tecnologia ActiveScripting:

```
'codice VBA
Dim oSC As Object 'lo Script Control
Set oSC =
        CreateObject("MSScriptControl.ScriptControl")
With oSC
    .AllowUI = True              'è permesso l'uso di
                          interfaccia utente negli script
    .Language = "VBScript" 'il linguaggio usato sarà
                             VBScript, alternativamente
                          'si sarebbe potuto usare
```

```
                    qualsiasi altro linguaggio
                        'di scripting installato
    .UseSafeSubset = False 'non impone nessuna
                     restrizione per la sicurezza
End With
'a questo punto aggiungiamo il codice delle classi
                                     VBScript che
'effettuano il proxy dei nodi XML come visto in
                                       precedenza
oSC.AddCode "<codice della classe>"
```

Fatto questo possiamo procedere all'istanziazione di queste classi VBScript e al loro uso all'interno di Visual Basic. Infatti, trattandosi di normali istanze di interfacce IDispatch per utilizzarle in Visual Basic è sufficiente ottenere il puntatore all'istanza della classe stessa e assegnarlo ad un normale Object o Variant. Questo è possibile sfruttando il potente metodo Eval dello ScriptControl (e dello scripting più in generale) che permetta di valutare, facendosi ritornare il risultato in forma di variant, una qualsiasi espressione:

```
'codice VBA
Dim oCRM As Object
Set oCRM = oSC.Eval ("New vCRM")
```

A questo punto possiamo accedere a oMyOrder direttamente da codice Visual Basic esattamente come abbiamo fatto all'interno di VBScript:

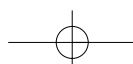
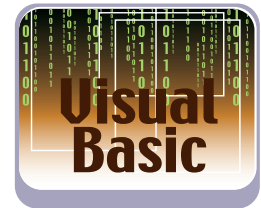
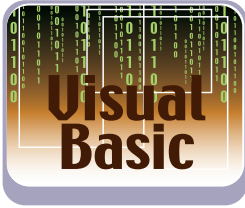
```
'codice VBA
oCRM.AUTHOR = "Vito Vessia"
Msgbox oCRM.AUTHOR
```

IL CLASS GENERATOR

Il meccanismo appena mostrato è molto potente ma ha, nella forma in cui l'abbiamo mostrato, un'utilità quasi nulla perché ci permette sì di scrivere le nostre classi di proxy intorno ad un qualsiasi XML con schema, ma ci costringe comunque a scriverci manualmente, e per ogni differente schema, il codice di tali classi. Ci costringe inoltre, quasi ancora più inutile, a istanziare tali classi e a popolarle secondo i dati presenti nel file XML in questione. A ben pensarci si tratterebbe di scrivere ogni volta sempre lo stesso codice perché gli schema XML si descrivono sempre nelle stesse modalità viste in precedenza e il popolamento delle istanze è ancora più banale perché non è altro che un susseguirsi interminabile di for each su nodi, sequence, ecc… Perché non affidarsi quindi ad un generatore automatico di classi a partire da schema e di istanze delle classi create in precedenza a partire da file XML che soggiacciono allo stesso schema? Ebbene è quello che andremo a realizzare, seppur in una forma non esaustiva.
Il sorgente che accompagna l'articolo presenta il progetto Visual Basic di un ActiveX Dll (CPXML Proxies.vbp) e di un progetto di esempio che utilizza tale engine (CPXML ProxyDemo.vbp) sul file XML mostrato nell'articolo. Il motore è però generico e funziona su ogni XML con schema standard **[3]** e utilizza il nuovo SOM di MSXML 4 per produrre le classi a partire dagli schema e il DOM per produrre le istanze delle classi a partire dai nodi XML. Questo presenta solo due metodi pubblici. Il primo, CreateClassesFromXML Schema, permette di produrre una gerarchia di classi VBScript a partire da uno schema XML:

```
'codice VBA
Public Function
  CreateClassesFromXMLSchema(XMLSchemaObject As
                                       Variant, _
                              Optional
          XMLSchemaRef As Variant) As String
```

Il primo parametro rappresenta proprio lo schema da cui produrre le classi. È di tipo variant perché potrebbe essere sia il nome del file o la url dello schema e sia direttamente l'oggetto ISchema ed infatti vengono fatti i controlli del caso (una IsObject e una QueryInterface) per comprenderne la natura. Il secondo parametro è passato invece per referenza e viene usato come parametro di tipo out, cioè viene valorizzato in uscita; in pratica a questo viene assegnata l'istanza dello schema appena prodotto. Il valore di ritorno della funzione è invece proprio il codice sorgente delle classi.
A questo punto si procede scorrendosi la collection elements dello ISchema, ciascun elemento rappresenta un diverso nodo dei file xml sottoposti allo schema dato. Vediamo un estratto della classe

```
Dim ISchemaItem As MSXML2.ISchemaItem
Dim IAttribute As MSXML2.ISchemaItem
Dim ISchema As MSXML2.ISchema
Dim IElem As MSXML2.ISchemaElement
Dim IComplex As MSXML2.ISchemaComplexType
Dim IProp As MSXML2.ISchemaAttribute
Dim IType As MSXML2.ISchemaType    For Each
                 ISchemaItem In ISchema.elements

CreateClassesFromXMLSchema =
                 CreateClassesFromXMLSchema & _
```

```
 "Class " & StrConv(ISchemaItem.Name,
                         vbProperCase) & vbCrLf

Set IElem = ISchemaItem
If IElem.itemType = SOMITEM_ELEMENT Then
   Set IComplex = IElem.Type
   CreateClassesFromXMLSchema =
         CreateClassesFromXMLSchema & vbCrLf & _
```

A questo punto, per ogni classe che si va a creare, si procede con l'enumerazione della collection Attributes del Complex ottenuto effettuando il casting dell'oggetto nodo (come si può osservare nel codice appena mostrato). Nel codice specifico vengono prodotti i membri privati della classe che conterranno i valori delle proprietà/attributi di ogni nodo.

```
For Each IAttribute In IComplex.Attributes
   CreateClassesFromXMLSchema =
         CreateClassesFromXMLSchema & vbCrLf & _
   "   Private m_" & IAttribute.Name
Next
```

Stessa cosa però sulla collection particles di contentModel del Complex. Essa conterrà l'elenco eventuale dei nodi figli del nodo corrente (ad es. il nodo vCRM ha come nodi figlia Order e Customer).

```
For Each IAttribute In
                 IComplex.contentModel.particles
    CreateClassesFromXMLSchema =
         CreateClassesFromXMLSchema & vbCrLf & _
   "   Public " & IAttribute.Name
Next
```

Un altro interessante pezzo di codice da osservare è quello che permette di gestire le constraint a livello di property let di ogni attributo, cioè di gestire correttamente le assegnazioni di valori agli attributi in base alle caratteristiche peculiari (tipo, lunghezza, ecc.) direttamente ereditate dalle informazioni di schema. Si tratta di un'implementazione molto parziale, ma rende l'idea sulla potenza del meccanismo che si va ad implementare: le classi che andremo a produrre non perderanno nulla delle caratteristiche dello schema XML di cui vanno a realizzare il wrapper.

```
For Each IAttribute In IComplex.Attributes
    CreateClassesFromXMLSchema =
         CreateClassesFromXMLSchema & vbCrLf & _
   "   Public Property Get " & IAttribute.Name &
                              "()" & vbCrLf & _
   "      " & IAttribute.Name & " = m_" &
                   IAttribute.Name & vbCrLf & _
   "   End Property" & vbCrLf & _
   "   " & vbCrLf & _
   "   Public Property Let " & IAttribute.Name &
             "(valueField)" & vbCrLf '& vbCrLf

   Set IProp = IAttribute
   Set IType = IProp.Type
   Select Case IType.itemType
      Case SOMITEM_DATATYPE_BYTE,
                    SOMITEM_DATATYPE_DECIMAL,
                   SOMITEM_DATATYPE_DOUBLE, _
         SOMITEM_DATATYPE_FLOAT,
                    SOMITEM_DATATYPE_INTEGER,
                       SOMITEM_DATATYPE_LONG,
         SOMITEM_DATATYPE_SHORT:
           CreateClassesFromXMLSchema =
                 CreateClassesFromXMLSchema & _
           "     If Not IsNumeric(0 & valueField)
                        Then " & vbCrLf & _
           "        Err.Raise 1970, , ""Invalid
                       number!""" & vbCrLf & _
           "     End If" & vbCrLf
      Case SOMITEM_DATATYPE_DATE,
                   SOMITEM_DATATYPE_DATETIME:
           CreateClassesFromXMLSchema =
                 CreateClassesFromXMLSchema & _
           "     If Not IsDate("""" & valueField)
                        Then " & vbCrLf & _
           "        Err.Raise 2000, , ""Invalid
                         date!""" & vbCrLf & _
           "     End If" & vbCrLf
   End Select
```

L'altro metodo è CreateInstanceFromXML. Esso banalmente scorre il file XML passato in input (XMLIssue)

```
Public Function CreateInstancesFromXML(XMLIssue
                              As Variant, _
                     Optional ParentArray As
                                Variant, _
                     Optional
                XMLSchemaRef As Variant, _
                     Optional SourceCode As
                   String = "") As Object
```

Il parametro XMLIsssue rappresenta il nome o il nodo XML che si vuole trasformare in un'istanza di una classe VBScript generata dallo schema a cui soggiace il nodo stesso. Vale infatti lo stesso discorso di ambivalenza visto per il parametro del metodo precedente. Gli altri parametri, invece, sono apparentemente poco significativi ed infatti non verranno mai adoperati (per questo sono opzionali) in modo diretto; servono invece all'interno del metodo stesso perché si tratta di un metodo ricorsivo. Infatti la logica del metodo

è che, dal nodo XML appena passato (o dal nodo root del file XML se il parametro XMLIssue è di tipo nodo) viene popolato un'istanza. Per ogni nodo figlio del nodo corrente viene invocato il metodo nuovamente per popolare le istanze di questi nodi figli e così via. I tre altri parametri servono proprio per comunicare informazioni alle altre istanze di ricorsione del metodo in modo da velocizzare alcune operazione e donare uno stato al metodo.

```
Dim lObject As Object
Dim lElement As MSXML2.IXMLDOMElement
Dim lDOM As MSXML2.DOMDocument40
Dim lXSDFile As String
Dim lAttribute As MSXML2.IXMLDOMAttribute
Dim lSchema As MSXML2.ISchema
Dim lSchemaItem As MSXML2.ISchemaItem
Dim lSchemaElement As MSXML2.ISchemaElement
Dim lSchemaComplex As
                    MSXML2.ISchemaComplexType
Dim lSchemaAttribute As MSXML2.ISchemaItem
Dim lNodeList As MSXML2.IXMLDOMNodeList
Dim lChildElement As MSXML2.IXMLDOMElement

If SourceCode = "" Then
   'ci troviamo al livello 0 della ricorsione e il sorgente
                              delle classi non è stato
   'ancora prodotto e quindi viene prodotto in questa
                         fase usando il metodo precedente
   '(codice omesso, per lo studio del codice di questo
                         metodo si rimanda ai sorgenti)
 Else 'è un livello di ricorsione >0 e quindi il sorgente
                           delle classi è stato passato
    'per referenza
    Set lSchema = XMLSchemaRef
End If

'nuova istanza della classe wrapper del nodo corrente
Set lObject = g_SC.Eval("new " &
                               lElement.nodeName)
```

Per ogni proprietà/attributo dell'istanza viene invocata la property let per assegnare il valore preso dal nodo XML originale per quell'attributo. La valorizzazione avviene per nome simbolico della proprietà grazie alla potente CallByName di Visual Basic:

```
For Each lSchemaAttribute In
                         lSchemaComplex.Attributes
    CallByName lObject, lSchemaAttribute.Name,
                                         VbLet, _
lElement.Attributes.getNamedItem(lSchemaAttribute.
                                  Name).nodeValue
    lObject.Attributes.Add lSchemaAttribute.Name, _
lElement.Attributes.getNamedItem(lSchemaAttribute.
                                  Name).nodeValue
Next
```

In questo brano di codice si può invece osservare l'uso della ricorsione nel metodo: per ogni nodo figlio di quello corrente viene invocato infatti lo stesso metodo.

```
For Each lSchemaAttribute In
              lSchemaComplex.contentModel.particles
    Set lNodeList =
   lElement.getElementsByTagName(lSchemaAttribute.
                                            Name)
    For Each lChildElement In lNodeList
        CreateInstancesFromXML lChildElement,
          CallByName(lObject, lChildElement.baseName,
                                         VbGet), _
          lSchema, SourceCode
    Next
Next
```

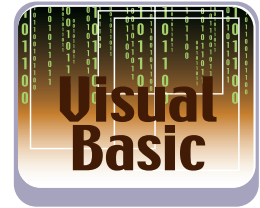
Vediamo dunque il motore appena creato in azione in un esempio molto semplice in Visual Basic:

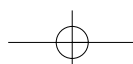
```
'codice VBA
Dim oProxy As Object
Dim oObject As Object

Set oProxy =
            CreateObject("CPXMLProxies.XMLProxy")
Set oObject =
             oProxy.CreateInstancesFromXML("dati.xml")
MsgBox oObject.Order(1).Row(2).Cover(1).CodQuest
```

Semplice e potente…

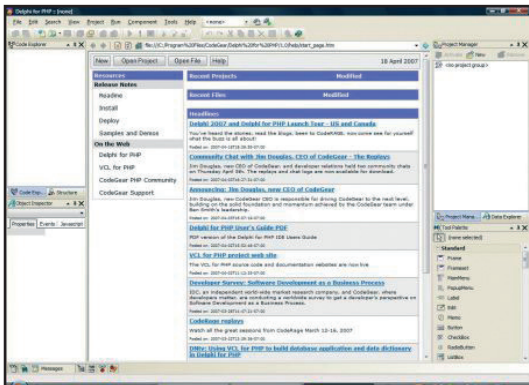
CONCLUSIONI

Abbiamo visto come realizzare un sistema di classi VBScript di wrapper a partire da un file XML con schema riciclando il concetto di classe/istanza sull'equivalente XML di schema/file. Le istanze di queste classi possono poi essere utilizzate a runtime dalle nostre applicazioni Visual Basic che quindi acquistano la capacità di gestire i complicati file XML con schema senza conoscere nulla di questo formato, senza ricorrere al potente ma un po' complicato object model di DOM e senza neppure mantenere una referenza a MSXML 4. Il tutto è stato generato facendo uso di un motore molto semplice scritto in Visual Basic che quindi ci mette in salvo dalla riscrittura di queste classi VBScript ogni volta che cambia il formato dello schema

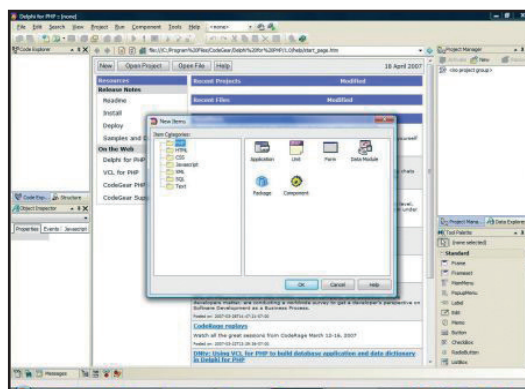
Vito Vessia

L'AUTORE

Vito Vessia progetta e sviluppa applicazioni e framework in .NET, Visual Basic/COM e Delphi occupandosi degli aspetti architetturali. Scrive da anni per le principali riviste italiane di programmazione ed è autore del libro "PROGRAMMARE IL CELLULARE", (Hoepli), 2002, sulla programmazione dei telefoni cellulari connessi al PC con protocollo standard AT+. Può essere contattato tramite e-mail all'indirizzo vvessia@katamail.com.

ECCO IL BORLAND DELPHI FOR PHP

MANCAVA A PHP UN'IDE RAD CHE CONSENTISSE DI PROGRAMMARE LE INTERFACCE GRAFICHE IN MODO VISUALE E CONTEMPORANEAMENTEDI GESTIRE EVENTI ED OGGETTI CON UNO STILE MODERNO. A RIEMPIRE IL GAP CI HA PENSATO BORLAND. ECCO COME...

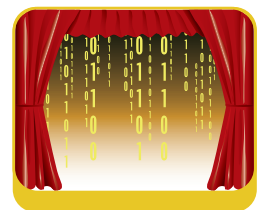
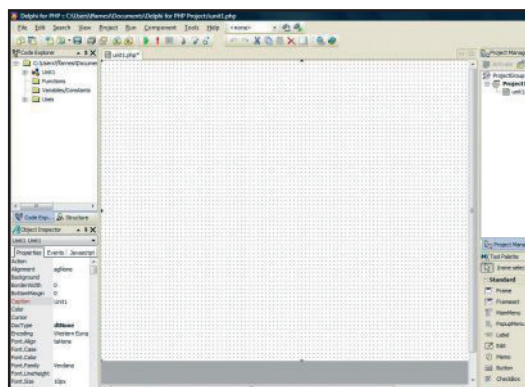
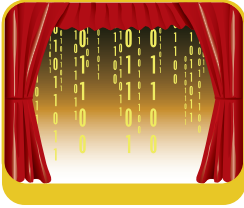
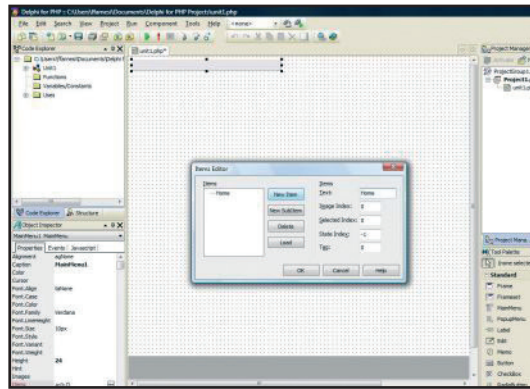
Ok, prima le buone notizie. Delphi per PHP è stato ufficialmente rilasciato e una versione trial di 14 giorni è disponibile per il download all'indirizzo http://www.codegear.com/Downloads/TrialandFreeVersions/Delphi/DelphiforPHP/tabid/250/Default.aspx. Oppure prelevarlo direttamente dal CD di ioProgrammo.

In tutti e due i casi per installarlo è necessario registrarsi al sito, riempire la form prima del download ed ottenere così una licenza valida. La licenza vi sarà inviata al vostro indirizzo email. Per attivare il prodotto è necessario copiare il file contenente la licenza nella vostra home di windows. Controllate la versione del vostro sistema operativo per sapere qual è la vostra home.

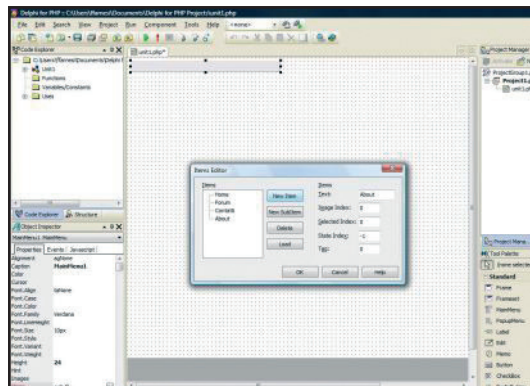
CHE COSA È?

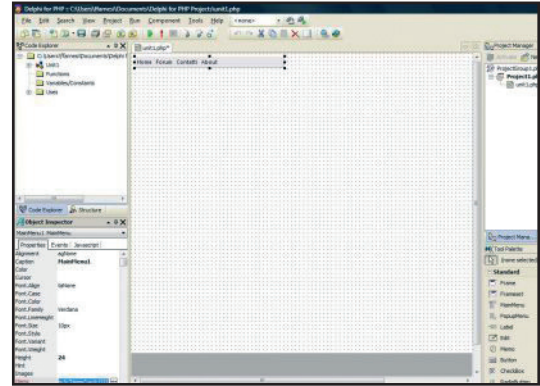
Si tratta di un editor RAD per PHP. Per RAD si intende un editor in classico stile Delphi o Visual Studio per intenderci. Strutturato in modo tale che esista un tavolo di disegno (la form) che rappresenta la pagina web e sul quale potete semplicemente trascinare con il mouse, da un'opportuna palette di componenti, i vari elementi che desiderate appaiano sulla vostra pagina.
Allo stesso modo ciascun componente è dotato di proprietà che ne definiscono ad esempio il colore, piuttosto che la forma o altre peculiarità del componente stesso. Ed infine ciascun componente può

reagire ad eventi specifici quali il doppio click o altri. Il vostro compito è semplicemente quello di programmare le azioni da compiere in relazione allo scatenarsi di un evento.

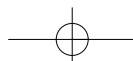
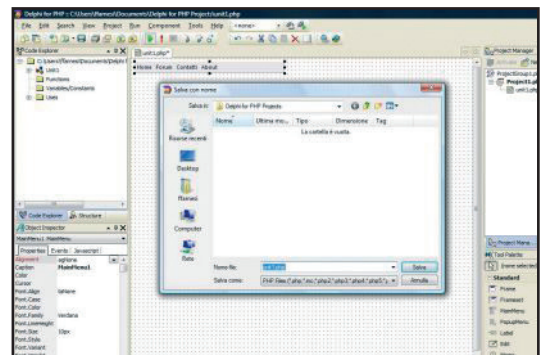
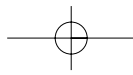
IN COSA È DIVERSO DAL SOLITO ?

I programmatori PHP non sono abituati ad avere a disposizione un editor visuale. Da sempre ciascun programmatore programma le proprie funzioni costruendo a mano le GUI, questo implica una profonda conoscenza di PHP. Ogni programmatore PHP a sua volta si autocostruisce un framework che conosce a menadito e che utilizza e modifica secondo le proprie esigenze. Con Delphi 4 PHP tutto cambia. La co-

noscenza di PHP fa da sfondo a quella che deve essere invece una conoscenza sostanziale del framework che fa da sostegno all'ambiente. In pratica non avrete bisogno di conoscere la funzione echo ma piuttosto la funzione che il framework vi mette a disposizione per settare una label o una caption.

LE COSE BUONE

Per chi ha un minimo di dimestichezza con gli ambienti RAD e soprattutto con quelli di Borland, utilizzare Delphi 4 PHP è un gioco da ragazzi. Per costruire ad esempio un menu a tendina è sufficiente trascinare l'apposito componente sulla form e settare un paio di proprietà. Allo stesso modo l'editor è dotato di code complexion, così come di sintax highlighting, di un debugger integrato e.. meraviglia delle meraviglie di un server web integrato per debuggare le vostre applicazioni. In pratica non dovete installare PHP sulla vostra macchina, non dovete installare Apache etc. Tutto quello che vi serve è già integrato in Delphi 4 PHP. La programmazione è quella classica RAD a componenti ed eventi. Si possono programmare gli eventi sia in JavaScript che direttamente in PHP.

IL FRAMEWORK SOTTOSTANTE

La parte più complicata di tutta la faccenda è comprendere come funziona il framework che sta alla base del prodotto. Non si tratta più di conoscere PHP se non nella sintassi base, ma conoscere il meccanismo e la logica di funzionamento del framework. Ad esempio per creare un semplice Hello World. ovvero progettare una pagina dotata di un bottone, una label e una textbox tale che al click sul bottone la caption della label cambi ricopiando il contenuto della textbox si possono seguire varie strade. Programmazione classica, utilizzando PHP, oppure gestione degli eventi tramite Javascript, oppure ancora Ajax. Tutti e tre i casi sono previsti dall'ambiente, tuttavia per ottenere il risultato voluto è necessario conoscere le funzioni esposte. L'Help in linea è comunque un ottimo aiuto.

CONCLUSIONI

Le prime impressioni sono molto positive. Delphi 4 PHP è una bella idea. Certo, è necessario stravolgere completamente le proprie abitudini programmative. Si deve conoscere profondamente il comportamento del framework ed abbandonare le convinzioni che ci siamo autoindotte in anni e anni di programmazione autoctona. Una volta fatto questo tuttavia l'innalzamento della produttività potrebbe essere enorme. Infine una nota sui componenti. Qualche anno fa quando Delphi fece la sua apparizione sul mercato, i componenti furono davvero una rivoluzione per l'informatica. Anche in questo caso è possibile sviluppare componenti. Avete bisogno di un bottone particolare? create il vostro componente e lo riutilizzate in tutti i vostri progetti. Molto molto comodo…

SOFTWARE SUL CD

Delphi for PHP

IL RAD PER PHP TARGATO BORLAND

PHP è uno dei linguaggi più utilizzati in programmazione, soprattutto per la sua vocazione verso il web. Le nuove versioni avevano risolto qualche problema legato alla mancanza di un forte orientamento alla programmazione ad oggetti, e dalla versione 5 in poi PHP è diventato davvero un linguaggio completo e professionale. Mancava un IDE RAD a PHP. A colmare questa lacuna ci ha pensato Borland con un prodotto il cui nome Delphi per PHP la dice lunga sulle caratteristiche dell'IDE. Sostanzialmente Delphi per PHP unisce ad un editore eccezionale un fortissimo orientamento visuale e ad eventi che consente di programmare interfacce sofisticate mediante semplice trascinamento dei componenti su una form e conseguente programmazione degli eventi ad essi associati. Si tratta di un modo nuovo di programmare applicazioni PHP che potenzialmente aumenta moltissimo la produttività di ogni programmatore. Si tratta ora di cambiare radicalmente il proprio modo di sviluppare per adattarsi alla logica contenuta nel framework sottostante l'IDE, ma una volta superato il primo impatto, programmare un'intera applicazione diventa davvero molto più semplice.

/Delphi4php/d4php_trial.exe

JRAT 1.0

OTTIMIZZIAMO IL CODICE

I profiler sono strumenti software in grado di analizzare il comportamento di un'applicazione a runtime ed estrarre utili informazioni quali il tempo d'esecuzione dei metodi, l'occupazione di memoria, ecc....Scopo di tali misurazioni è quello di identificare i colli di bottiglia dell'applicazione per eseguirne il refactoring e ridurre così il tempo d'esecuzione a tutto vantaggio delle prestazioni. Il profile che qui vi presentiamo è JRAT, pensato per Java ed OpenSource Jrat instrumenta le classi dell'applicazione in modo che ogni metodo eal principio e al termine delle sua esecuzione scriva su un log dei riferimenti temporali. L'interfaccia grafica di Jrat permette poi di visualizzare tale log sottoforma di un albero dove ogni nodo rappresenta la chiamata ad un metodo.

/jrat/shiftone-jrat.jar

STRUTS 2.0.1

IL FRAMEWORK MVC PER JAVA

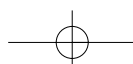
Il pattern MVC è probabilmente il piùusato da tutti gli sviluppatori in ogni linguaggio.Questo framework è il leader per quanto riguarda lo sviluppo di applicazioni Java secondo il pattern MVC. Molto comodo, stabile e affidabile, rappresenta la base di partenza per applicazioni che devono essere facilmente manutenibili. Uno standard da usare se progettate applicazioni di dimensioni generose, ma anche quando si sviluppano web application professionali

struts-2.0.1-all.zip

SPRING 2.0.2

IL FRAMEWORK DEI FRAMEWORK

Spring è un Framework che riuniscesotto un unico cappello una serie di strumenti già esistenti mettendoli in grado di comunicare in modo correttofra loro. Inoltre Spring implementa il pattern Ioc, inversion of control che garantisce un alto grado di disaccoppiamento e una maggiore manutebilità del codice

spring-framework-2.0.2-with-dependencies

ECHO 2 2

AJAX SEMPLICE ANCHE CON JAVA

Ajax è ormai il re incontrastato del web. Ma come scrivere applicazioni che fanno uso di Ajax anche in Java? La risposta è semplice: utilizzando JavaScript. Ma se si vuole sviluppare in modo rapido e sicuro è sempre meglio fare uso di un buon framework. Echo2 è un framework per lo sviluppo di applicazioni Ajax in tecnologia Java. Si tratta di un framework completamente opensource a sua volta scritto interamente in java, perciò molto personalizzabile

/echo2/NextApp_Echo2.zip

EMULE SRC 1.0

I SORGENTI DI EMULE!

In questo numero di ioProgrammo abbiamo analizzato il comportamento di una rete PHP, seguendo passo passo il diagramma di flusso del programma più conosciuto in questo ambito: eMule.Qui vi presentiamo i sorgenti del programma, che potete analizzare per realizzare un vostro client oppure personalizzare secondo le vostre esigenze. Si tratta di un ottimo esempio di programmazione, che consente a molti di apprendere i principi base del funzionamento di una rete P2P.

/emulesrc/eMule0.47c-Sources.zip

REACTOS SRC 0.3.1

IL CODICE DI WINDOWS

ReactOS è un progetto OpenSource che mira ad eseguire un clone pienamente funzionante del sistema operativo Windows.Il progetto è già in fase avanzata e ne esistono già dei binari. In questo numero di ioProgrammo vi presentiamo però i sorgenti che stanno alla base del kernel del sistema operativo. Si tratta di un ottimo modo per imparare come si programma a basso livello la gestione della memoria, dei processi e la gestione di un file system. Realmente un bel progetto che vale la pena di essere studiato.

/reactosrc/ReactOS-0.3.1-REL-src.zip

IDA PRO DISASSEMBLER 5.1

Il debugger che mette a nudo il software

Quante volte vi è capitato di dover scoprire il comportamento di un programma senza disporre dei sorgenti? Quante volte vi è capitato di dover continuare un progetto iniziato da altri e di non disporre del codice? In questi casi quello che ci vuole è un buon disassembler che consenta di tracciare il comportamento di un eseguibile riportandolo sotto forma di codice. IDA è uno dei migliori, dispojne di una sofisticata interfaccia grafica ed è in grado di eseguire analisi sofisticate.

/idademo/idademo51.exe

WINSCP 4.00

CONNESSIONI IN SICUREZZA

Dovete uploadare un file sul vostro sistema di hosting e l'unico protocollo supportato è ssh? La soluzione a questo problema esiste e si chiama SCP. Peccato che volendolo usare in modo tradizionale sia necessario utilizzare una noiosissima riga di comando. Ed ecco che ci viene in aiuto WinScp che con una semplice interfaccia consente di gestire i file in remoto anche usando un protocollo sicuro come SSH.

/winscp/winscp400setup.exe

JAVA SE DEVELOPMENT KIT 6

IL COMPILATORE INDISPENSABILE PER PROGRAMMARE IN JAVA

Se avete intenzione di iniziare a programmare in Java oppure siete già dei programmatori esperti avete bisogno sicuramente del compilatore e delle librerie Java indispensabili. Sotto il nome di Java SE Development Kit vanno appunto tutti gli strumenti e le librerie nonché le utility necessarie per programmare in JAVA. L'attuale versione è la 6.0, ovvero la nuovissima release densa di innovazioni e molto più legata al desktop di quanto non fossero tutte le precedenti

jdk-6-windows-i586

ECLIPSE SDK 3.2.2

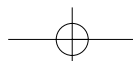
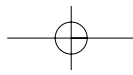
L'IDE TUTTOFARE

Eclipse è un progetto completo portato avanti da Eclipse Foundation con la collaborazione di una miriade di aziende fra cui IBM, Adobe, Sun e che si è prefissata lo scopo di creare un IDE estendibile per plugin adattabile a qualunque tipo di linguaggio o tecnologia. Di default Eclipse si propone come IDE per Java ed è qui che da il meglio di se. Ma proprio grazie ai suoi plugin è possibile utilizzarlo come ambiente di programmazione per PHP, per C++, per Flex e per molti altri linguaggi ancora. Inoltre sempre grazie per ciascun linguaggio sono disponibili altri plugin ad esempio per rendere l'ambiente RAD o per favorire lo sviluppo dei Web Services o altro. Insomma lo scopo è stato raggiunto completamente. Eclipse è realmente un IDE tuttofare, ormai maturo, e che serve una miriade di programmatori grazie alle sue caratteristiche di affidabilità e flessibilità. Unica nota negativa: una certa pesantezza che lo rende idoneo ad essere usato solo su PC con una dotazione hardware minima di tutto rispetto

eclipse-SDK-3.2.1-win32.zip

PHP 5.2.1

IL LINGUAGGIO DI SCRIPTING PIÙ AMATO DEL WEB

Sono tre le colonne portanti di Internet: PHP, APACHE e MySQL. Certo la concorrenza è forte. Asp.NET e SQL Server avanzano con celerità, ma a tutt'oggi non si può affermare che i siti sviluppati in PHP costituiscano la stragrande maggioranza di Internet. Quali sono le ragioni del successo di cotanto linguaggio? Prima di tutto la completezza. PHP ha di base tutto quello che serve ad un buon programmatore, raramente è necessario ricorrere a librerie esterne, e quando è proprio indispensabile farlo esistono comunque una serie di repository che rendono tutto immediatamente disponibile ed in forma gratuita. Il secondo punto di forza del linguaggio sta nella sua capacità di poter essere utilizzato sia in modo procedurale che nella sua forma ad oggetti certamente più potente e completa. Esiste un terzo

di punta di forza essenziale che è quello riguardante la curva di apprendimento. PHP è in assoluto uno dei linguaggi con la curva di apprendimento più bassa nel panorama degli strumenti di programmazione. Si tratta perciò di uno strumento indispensabile per chi si avvvicina alla programmazione web, a meno che non intendiate scegliere strade diverse quali possono essere ASP.NET o JSP

php-5.2.0-Win32.zip

PHPMYADMIN 2.10 RC1

IL GESTORE DI MYSQL

MySQL è probabilmente il database più utilizzato per le applicazioni web. Ma come gestirne gli utenti? Come creare un nuovo database? Quale interfaccia utilizzare per accedere e gestire i dati? A tutti questi problemi ed altri ancora risponde PHPMyAdmin, probabilmente la più sofisticata interfaccia di gestione per MySQL. Non solo, ma PhpMyadmin è un'applicazione web, che può facilmente essere utilizzata anche in modo remoto e su sistemi in hosting. Praticamente un must per chi programma applicazioni PHP che fanno uso di database.

/phpmyadmin/phpMyAdmin-2.10.1-rc1-all-languages.zip

MYSQL 5.1.15

IL PRINCIPE DEI DATABASE

Indispensabile per programmare web application in tecnologia PHP. Nonche non sia possibile utilizzare altridatabase, ma MySQL e PHP rappresentano veramente un binomio inscindibile.
L'integrazione fra questo database e il linguaggio di scripting più usato sulla rete è talmente alta da fare divenire quasi un obbligo l'uso congiunto di questi due strumenti

mysql-5.0.27-win32.zip

FCKEDITOR 2.4

L'EDITOR ONLINE

Volete inserire un testo in un database? Vi piacerebbe che fosse già formattato in maniera particolare?Le soluzioni sono 2, o seguite il nostro articolo su come si sviluppa un vostro editor personalizzato, oppure inserite questo meraviglioso controllo nella vostra applicazione. FCKEditor è un editor completo, che consente di utilizzare tutte le formattazioni tipiche di un word processor evoluto e che agevola di molto il compito del programmatore

/fckeditor

WORDPRESS 2.1.3

IL RE DEI BLOG

Wordpress è probabilmente lo strumento più diffuso per la programmazione di un vostro blog.Si tratta di uno strumento particolarmente leggero e semplice da usare. Nonostante questo l'architettura a plugin lo rende particolarmente estendibile. Molto interessante la struttura delle API che con una serie di hook rende la programmazione dei plugin un'attività relativamente semplice da svolgere

/wordpress/wordpress-2.1.3.zip

DOJIO 0.4.2

IL NUOVO FRAMEWORK PER JAVASCRIPT

JavaScript dopo un periodo di appannamento sembra essere tornato alla ribalta sulla scena della programmazione per il web da quando Ajax è diventato l'argomento principe per lo sviluppo del Web 2.0.Proprio JavaScript infatti rappresenta il cuore di Ajax.Questa rinnovata popolarità ha dato il via alla nascita di una serie di framework che facilitano la programmazione JavaScript. Fra i tanti spicca dojo. Prima di tutto per la sua innata propensione alla programmazione ad eventi, ma anche e soprattutto perché consente in una qualche misura di utilizzare la tecnologia ajax in push invece che in pop. Ovvero in modo tale che sia il server ad inviare i dati al client e non viceversa. In realtà questa possibilità e in via di sviluppo e appare soltanto come una delle tante tecnologie che dojo offre, ma accanto a tutte le agevolazioni contenute in dojo, quest'ultima non è da sottovalutare

/dojo/dojo-0.4.2-ajax.tar.gz

JOOMLA 1.0.12

L'EREDE DI MAMBO

Un cms dalle potenzialità straordinarie che offre una serie di funzionalità decisamente evolute. Dalla personalizzazione di una singola pagina fino ad un'architettura a plugin molto sofisticata che virtualmente consente di aggiungere al framework ogni tipo di funzionalità

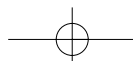
/joomla/Joomla_1.0.12-Stable-Full_Package.zip

DRUPAL 5.1

IL COSTRUTTORE DI BLOG

Drupal è prima di tutto un sistema CMS, consente cioè la pubblicazione dei contenuti in modo semplificato, di modo che anche gli utenti meno smaliziati possano inserire dei contenuti su Internet pur non avendo conoscenze di programmazione o di html, ma Drupal è anche molto di più. Utilizzando Drupal ognuno degli

utenti iscritti al sito diviene proprietario di un blog e con pochi accorgimenti potrete fare in modo che ciascuno di questi blog goda di una vita propria indipendente dal sito principale, oppure se lo desiderate potete far comunicare i vari blog in modo da creare una vera community di bloggers

drupal-5.1.tar.gz

CALCOLO DELLA MEDIANA

FIND È UN ALGORITMO PER IL CALCOLO DELLA MEDIANA. SI TRATTA DELLA BASE TEORICA DEL CONOSCIUTO QUICK SORT, NON SEMPRE USATO PER IL SUO SCOPO, APPUNTO CALCOLARE IN MODO EFFICIENTE UN VALORE MEDIO

Inauguriamo un nuovo filone di analisi e approfondimento. Tratteremo soluzioni per ben definiti problemi che si possono incontrare nel nostro lavoro di programmatori. Avremo modo di studiare e approfondire quesiti afferenti alla matematica e alla fisica, così come problemi di statistica e ottimizzazione. Ma saranno oggetto delle nostre esplorazioni anche problemi difficilmente classificabili ma che spesso ci capita di incontrare, come ad esempio la produzione di un calendario di incontri per abbinare le varie squadre nelle varie giornate o la valutazione della dimensione di una popolazione.
Il problema che ci proponiamo di affrontare e risolvere è di statistica ed è il calcolo della mediana di una serie di numeri. Utilizzeremo per la soluzione il metodo proposto da C.A.R Hoare. L'algoritmo in questione rappresenta le fondamenta del conosciuto quick sort. Non a caso Find è anche conosciuto come quickselect o one sided quicksort.

IL CALCOLO DELLA MEDIANA

In molte applicazioni, in svariati campi dall'automazione industriale alla metereologia, è richiesto il calcolo della mediana. Si tratta di un valore medio come la media aritmetica, la media geometrica, la media armonica o la moda che sintetizza in un unico valore una serie di numeri. In particolare, per la serie di numeri ordinata non decrescente la mediana è esattamente l'elemento centrale. Se la cardinalità delle serie è dispari l'elemento centrale è unico. Qualora la serie è costituita da un numero pari di elementi da un punto di vista prettamente teorico si assume come mediana la semisomma dei due elementi centrali. Nella pratica, in questa seconda eventualità, si può considerare anche uno dei due valori centrali. Tra i valori medi la mediana, per come è stata definita, è tra i più ostici da calcolare poiché, contrariamente agli altri, presuppone l'ordinamento dei dati da trattare. Faccio una piccola anticipazione. L'algoritmo find sottende una brillante idea che rende evitabile il processo di ordinamento e abbatte quindi la complessità computazionale. Supponiamo per semplicità che il numero n dei dati sia dispari, per evitare il calcolo di semisomme, del resto estendere il calcolo a n pari non comporterebbe particolari sforzi implementativi. Se V è un vettore di n elementi il modo tradizionale per calcolare la media è ordinare il vettore in modo che risulti

V[1]<=V[2]<=…<=V[m]<= … V[n-1]<=V[n]
rel.(1)

Dove m è esattamente l'indice centrale m=(n+1)/2 e conseguentemente V[m] l'elemento centrale, ossia la mediana. Facciamo subito un esempio per intenderci. Supponiamo che un vettore *temper* contenga le temperature registrate a intervalli orari in un determinato luogo, si suppone anche che le temperature siano rappresentate con numeri interi. Sia temper il seguente vettore:

15	14	12	11	14	17	18	19	20	19	16

Se il vettore viene ordinato in senso non decrescente si ottiene:

11	12	14	14	15	16	17	18	19	19	20

VALORI MEDI

Esistono diversi valori medi esaminiamone alcuni diversi dalla mediana. Il più conosciuto e usato è la media aritmetica. È quel valore che sostituito ai dati in esame ne lascia invariata la somma. Si calcola quindi sommando tutta la serie dei numeri e dividendola per il numero di numeri. La media geometrica di n valori è un numero g che rispetto ai valori stessi lascia invariato il loro prodotto. Si utilizza quando ha senso moltiplicare dati statistici.
La media armonica è uguale al reciproco della media aritmetica dei reciproci. Si calcola quando ha senso calcolare i reciproci.
La moda o valore normale è il valore al quale corrisponde la massima frequenza. Detto diversamente è il valore che si ripete più volte.

Nell'esempio n è 11 e m è 6. La mediana risulta il sesto valore del vettore ordinata ossia 16. Ovviamente se avessimo considerato il valore centrale prima del processo di ordinamento, avremmo commesso un errore

ALGORITMO FIND

L'algoritmo messo a punto da Hoare per il calcolo della mediana si basa su una semplice quanto astuta idea, non necessariamente si deve ordinare il vettore, è sufficiente individuare un numero della serie per il quale la metà dei valori sia minore e l'altra meta sia maggiore. Quindi è sufficiente "disporre" opportunamente i dati nel vettore in modo che siano verificate semplici proprietà. Prima di analizzare le proprietà è utile sottolineare che si tratta di disporre in modo generalmente diverso da quello iniziale i dati e non ordinare. In alcuni testi si trova in modo equivoco il termine ordinare, anche se non si tratta del tradizionale processo di ordinamento che conosciamo con cui si dispongono i dati secondo al relazione 1. Le proprietà possono essere così sintetizzate:

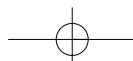
1. L'elemento in posizione centrale nel vettore deve essere centrale in ordine di grandezza rispetto agli altri;
2. Gli elementi che precedono quello centrale siano minori di esso;
3. Gli elementi che seguono quello centrale siano maggiori di esso;

Tali proprietà si possono esprimere con la seguente relazione.

V[1],V[2]…V[m-1]<=V[m]<=V[m+1] …V[n-1],V[n]
rel.(2)

Qui V[m] è la mediana. Certamente si tratta di una relazione meno "forte" della prima (1) che prevedeva l'ordinamento completo dei dati, ad ogni modo è sufficiente per il calcolo della mediana. Se, inoltre, garantirà una complessità minore dell'ordinamento, sarà da preferire ad esso nel caso in cui si voglia soltanto calcolare la mediana. L'algoritmo find, come si poteva prevedere, si basa proprio sulla disposizione dei dati in un vettore in modo che assumano la relazione 2. Analizziamone quindi il comportamento. Si comincia percorrendo il vettore V, ad esempio, da sinistra (non ha importanza il punto di partenza) verso destra. Ogni elemento scandito di valore "basso" deve essere lasciato nella sua posizione mentre ogni elemento "alto" deve essere spostato nell'estremità destra scambiandolo con un corrispettivo basso. Un primo problema è individuare il secondo elemento da scambiare. Si risolve facendo un'esplorazione a ritroso, da destra verso sinistra, fino a quando non si incontra un valore basso. Fatto lo scambio di fatto si continuano a percorrere in direzione opposta i due percorsi: da sinistra verso destra e da destra verso sinistra alla ricerca rispettivamente di elementi alti e bassi utili ad essere scambiati tra loro. Il processo si interrompe quando i due cammini si incontrano in un punto intermedio del percorso (del vettore). Non a caso la prima volta che ho citato i valori bassi e quelli alti li ho virgolettati; si rende necessario un criterio per definirli. Bisogna considerare un valore di confronto rispetto al quale classificarli, un pivot che si comporti da spartiacque, in modo che se il valore del vettore è più piccolo del valore di confronto si considera basso, altrimenti si considera alto. La scelta ricade su un valore presente nella serie. Un criterio di scelta può essere proprio il valore V[m]. Come vedremo nel prossimo paragrafo tale scelta è proprio una discriminante rispetto alla bontà dell'algoritmo, e molti studi sono stati fatti a riguardo. Alla fine della scansione l'incontro tra i due cammini definirà un elemento di separazione, se questo occupa un posto a destra del centro, un numero p maggiore di m, allora la mediana dovrà trovarsi a sinistra della separazione altrimenti se p è minore di m si troverà certamente a destra. Nel primo caso gli elementi a destra della separazione saranno al passo successivo ignorati; nel secondo caso ad essere ignorati saranno i valori a sinistra della separazione. Si innescano due nuovi cammini uno dei quali questa volta avrà inizio dall'elemento di separazione anziché dall'estremo del vettore come nel primo passo. Verrà stabilito un nuovo elemento di confronto e quando nuovamente i due cammini si incontrano terminerà il secondo passo. Anche in questo caso il punto di incontro porta essere a sinistra o a destra di m. In entrambi i casi si ragionerà come abbiamo fatto in precedenza, ignorando la parte sinistra se il nuovo p è minore di m o la parte destra altrimenti. Il processo terminerà quando il valore p risultante, ossia l'indice di separazione dei due tronconi di vettore (sottovettori ad ogni iterazio-

COME SCEGLIERE I VALORI MEDI

Ogni valore medio viene usato in base alle sue caratteristiche e proprietà. Ad esempio la proprietà della media aritmetica per la quale la somma degli scarti tra i valori e la media stessa dia zero la rende largamente utilizzata. Per la mediana si ha che la somma dei valori assoluti degli scarti tra i singoli valori e la mediana è un minimo rispetto ad ogni altro valore preso come riferimento per il calcolo degli scarti. La media armonica ad esempio è usata per il calcolo del potere di acquisto che è definito come la quantità di merce acquistabile con una data unità di moneta. In questo caso è utile ricordare che il potere di acquisto è il reciproco del prezzo della merce, per questo è utile usare la media armonica.

ne sempre più piccoli) risulterà uguale a m, ossia all'indice centrale. In tale situazione sapremo che è rispettata la relazione 2 che decreta V[m] come mediana.
Sviluppiamo un esempio su un caso concreto per capire meglio il meccanismo. Consideriamo il se-

10	2	6	9	4	5	12	1	7

guente vettore di 9 elementi.

Si tratta di un vettore sicuramente disordinato e che non rispetta neanche la relazione 2. Applicando l'algoritmo find intendiamo modificarlo facendo in modo che rispetti proprio la relazione 2 e che quindi l'elemento centrale, in posizione 4 (considerando la prima cella in posizione zero e l'ultima in posizione 8, in accordo con il codice C++ sviluppato nel prossimo paragrafo) ci sia la mediana. Si stabilisce inizialmente un valore di confronto che risulta essere quello centrale, ossia 4. A tale proposito è utile ricordare che la scelta del pivot così fatta equivale ad una scelta puramente casuale visto che in posizione centrale potrebbe trovarsi nel caso peggiore il valore più grande o più piccolo che risulterebbero poco utili. Si parte da sinistra alla ricerca di valori alti da scambiare con valori bassi ottenuti dal percorso contrario a partire da destra. Il primo valore 10 è già alto poiché più grande di 4. Nel senso opposto il primo valore 7 non è basso come lo si desidera essendo maggiore di quattro, mentre il secondo incontrato 1 è basso, infatti è minore di 4. I due valori si possono scam-

1	2	6	9	4	5	12	10	7

biare. Si ottiene:

Continuando la scansione da sinistra si incontra 2 che non è alto e 6 che invece è alto (>4). Da destra la ricerca scarta 12 e 5 entrambi alti, mentre trova proprio il valore di confronto 4 essendo non maggiore di se stesso. Nel caso specifico v[4]=4 è sia valore di confronto che elemento del vettore sottoposto a ricerca di valori bassi. Ne risulta che 4

1	2	4	9	6	5	12	10	7

viene scambiato con 6. Il nuovo vettore è:

L'ulteriore scorrimento da sinistra a destra si individua un nuovo elemento alto 9>4, purtroppo i due cammini si sono adesso incontrati e quindi la prima fase ha termine. Si individua come elemento di separazione proprio 9 che purtroppo non è posizionato al centro. Il fatto che l'elemento di separazione cada nella prima sinistra del vettore ci suggerisce che i valori a sinistra del 9 dovranno essere ignorati nella nuova fase poiché risultano già definitivamente disposti. Da sinistra partirà quindi proprio dal numero 9 mentre da destra da 7. Il nuovo valore di confronto è scelto con il solito criterio come quello centrale ed è quindi 6. Rispetto a 6, il numero 9 risulta alto. Nel cammino in senso opposto bisogna trovare un basso con cui scambiarlo, si scarta oltre al 7 anche il 10 e il 12; è valido questa volta il 5 che rispetto a 6 è basso. Si scambiano

1	2	4	5	6	9	12	10	7

quindi 9 e 5 è si ottiene un nuovo vettore.

La scansione da sinistra si ferma su 6 che esattamente l'elemento centrale. L'algoritmo si può fermare. Si sono rese necessarie due sole fasi. L'elemento al centro il numero 6 è la mediana e il vettore rispetta la relazione 2. Si veda la figura 1 che testa lo stesso vettore sul codice sviluppato. Solo per una casualità la prima parte del vettore è anche ordinata, come si vede la seconda parte non è ordinata, semplicemente rispetta la condizione imposta per il calcolo della mediana.

IMPLEMENTAZIONE

La funzione find() avrà il compito di disporre in modo opportuno un vettore v di numeri interi in modo che sia verificata la relazione 2 secondo l'algoritmo descritto nel paragrafo precedente. Si focalizza l'attenzione su una porzione di vettore sempre più piccola i cui estremi man mano convergono verso il centro. Utilizzeremo due variabili iniz e fin per mantenere traccia di tali estremi. Altre due variabili chiamate sx e dx verranno invece usate per effettuare i cammini da sinistra a destra e da destra a sinistra. L'elemento di separazione sarà contenuto nella variabile con. Ovviamente con m indicheremo l'indice centrale. Ecco il codice C++ per la funzione vuota find().

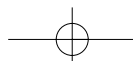
```
// funzione per il calcolo della mediana
void find()
{  int m,iniz,fin,sx,dx,con,temp;
   iniz=0;
   fin=n-1;
   m=(iniz+fin)/2;
   while (iniz<fin)
   {   con=v[m];
       sx=iniz;
       dx=fin;
       while (sx<=dx)
       { // scorre verso  destra
          while (v[sx]<con) sx++;
          // scorre verso  sinistra
          while (con<v[dx]) dx--;
          if (sx<=dx)
          {  temp=v[sx]; //scambio
            v[sx]=v[dx];
            v[dx]=temp;
            sx++; dx--;
          }
```

```
C:\aa\edmaster\ioprogrammo\n_115\so
n -> 9
v [0] -> 10
v [1] -> 2
v [2] -> 6
v [3] -> 9
v [4] -> 4
v [5] -> 5
v [6] -> 12
v [7] -> 1
v [8] -> 7

Mediana -> 6

Vettore trattato

v [0] ->        1
v [1] ->        2
v [2] ->        4
v [3] ->        5
v [4] ->        6
v [5] ->        9
v [6] ->        12
v [7] ->        10
v [8] ->        7
```

Fig. 1: "Programma find in esecuzione"

```
        };
        if (m<=dx) fin=dx;
        else if (sx<=m) iniz=sx;
            else iniz=n; // forza l'uscita
    };
    cout<<"\nMediana -> "<<v[m]<<"\n";
}
```

Il ciclo principale è attivo intanto che iniz risulta minore di *find*. All'interno del ciclo si stabilisce il valore di separazione e si inizializzano i due contatori per scorrere nei due sensi nel vettore. Un ciclo innestato permane fin quando i due contatori sx e dx appena citati non si incrociano. Nel ciclo si dovrà prima scorrere verso destra con la variabile sx e poi verso sinistra con dx. Quando queste due scansioni realizzate con altrettanti cicli danno esito positivo vuol dire che sono stati individuati i valori "alti" e "bassi" rispetto al separatore; è possibile quindi scambiarli. Da notare che i due indici scorrono se i valori corrispondenti di sx e dx sono rispettivamente strettamente minori e maggiori del separatore. Quindi l'eguaglianza con il separatore ne sancisce un elemento opportuno per lo scambio. Alla fine di queste operazioni resta da stabilire se aggiornare fin o iniz che equivale come spiegato nella parte descrittiva a trascurare la parte iniziale o finale del vettore (o più in generale del sottovettore). In figura 1 è riportata una cattura del programma che mostra simultaneamente l'input fornito e l'output ottenuto.

COME SCEGLIERE I VALORI MEDI

Il quick sort come il nome fa intuire è un algoritmo di ordinamento e si basa sul concetto su cui si regge find. Si separano sottovettore rispetto a un pivot e si ripete ricorsivamente il processo per tutti gli spezzoni di vettore che si creano dai conseguenti partizionamenti. Nella figura riportata di seguito si possono osservare le varie fasi che portano all'ordinamento.
Data la natura ricorsiva il quick sort può essere espresso in modo naturale e sintetico. Il record della sintesi si raggiunge in Ruby. Ecco come si realizza quick sort

```
def qsort(list)
  return [] if list.size == 0
  x, *xs = *list
  less, more = xs.partition{|y| y < x}
  qsort(less) + [x] + qsort(more)
end
```

Fasi\indici	1	2	3	4	5	6	7	8	9	Pivot
0	15	12	36	9	29	7	18	30	10	29
1	15	12	10	9	18	7	29	30	36	10 e 30
2	7	9	10	12	18	15	29	30	36	7 e 18
3	7	9	10	12	15	18	29	30	36	12
4	7	9	10	12	15	18	29	30	36	

"Fasi del quick sort. Per ogni fase sono specificati gli elementi di pivot. Il colore verde indica le parti del vettore parzialmente ordinate. Nell'ultima fase l'intero vettore è ordinato".

APPROFONDIMENTI

L'analisi della complessità per find non è molto semplice da attuare. Fermo restando che si hanno performance sicuramente migliori al caso del completo ordinamento, nel caso di find pesano come macigni i rari ma presenti casi peggiori. Nella situazione media è stato stimato una complessità lineare visto che si tratta di scandire un numero circoscritto di volte il vettore, anzi porzioni sempre più piccole di vettore. Esistono sostanziali miglioramenti dell'algoritmo tra questi è utile citare Schonhage e LazySelect accreditato con complessità 3n nel caso peggiore.
Alcune accortezze sono state messe a punto per migliorare le prestazioni. Un esempio per tutti: il valore di confronto può essere scelto come la mediana di tre valori scelti casualmente, questo scongiura l'ipotesi di scegliere in prima istanza il valore più grande o più piccolo, con un trascurabile aggravio computazionale.

CONCLUSIONI

L'esempio trattato in questo numero ribadisce ancora una volta una delle esigenze del programmatore, ovvero ottimizzare qualsiasi processo. Per cui se per calcolare la mediana si può risparmiare del tempo siamo disposti a utilizzare un algoritmo più efficiente. In molti ambiti, soprattutto quello industriale e del controllo, tali differenze non sono da poco. Vi aspetto alle prossime puntate per esplorare altri argomenti dello stesso tipo.

Fabio Grimaldi